

ANOTHER ONES BITES THE DUST!

Diesmal ist es das WoMo-Team!

Und vielleicht auch der SPC... s. Seite 2

Wolfgang Haller, Telefon 0221/680 33 10
Im Tannenforst 10, 51069 Köln
E-mail: womoteam@t-online.de
Homepage: http://www.womoteam.de
Bank: DVB, BLZ 370 604 26, Konto-Nr. 7404 172 012

Ausgabe 139/140

Juli/August 2001

NoMore WoMo-Team

Einige wissen es schon, viele aber werden es erst jetzt an dieser Stelle erfahren: Das WoMo-Team, welches 1990 diesen Club hier übernommen und etwas fast totgeglaubtes bis zum heutigen Tage aufrecht halten konnte, wird es in Zukunft nicht mehr geben.
Ich habe mich mit der Fertigstellung dieses Infos wirklich schwergetan. Die Qualität läßt sicher einiges zu wünschen übrig, der Inhalt ist aber sicherlich wieder für alle interessant. Ich werde auch die restlichen Ausgaben dieses Jahres noch hinbekommen. Für das kommende Jahr hingegen kann und mag ich zu diesem Zeitpunkt keine Prognose stellen. Ich werde in der letzten Ausgabe 2001 ganz konkret schreiben, ob ich weitermache oder nicht. An alle, die im Laufe der Jahre persönlich mit uns befreundet waren habe ich noch eine Bitte: Haltet uns als das alte WoMo-Team in guter Erinnerung. Es gibt auch keinen Grund, jeglichen Kontakt abzubrechen, egal ob der Club bestehen bleibt oder nicht.
Wer mich persönlich sprechen will, der sollte zukünftig die Telefonnummer:

0221/680 33 10

anwählen. Es geht aber auch nach wie vor unter der alten 0221/68 59 46.
Danke an euch alle für euer Verständnis, Wo.

This is an explanation for our readers in english.

NoMore WoMo-Team

It was in January 1990, when a new „group" started to take over a so called sinking ship: The Spectrum Profi Club. Because many people guessed the Sinclair Computers in agony. This group was very fast well known in the Spectrum scene as WoMo-Team, created by the first letters of the prenames from the formers Wolfgang and Monika.
In the years below 1990 we become more and more an international club with a lot of new contacts and friends. With a lot of foreign clubs we started a good communication with experience- and software-exchange.
Now, 10 and a half years later I feel sorry to tell you, that this team no longer exists because of separation.
I beg your pardon if this mag lacks in the usual quality but it was very hard for me to finish it. However, I did it and i think the content is still interesting for all of you.
What will go on with this club? To be true, I can´t say it yet. I think I can promise, that you got this mag until the end of this year, but I have no idea for 2002.
More to this point in the last issue this year. Keep that old WoMo-Team in good remembrance.
Thank´s for your comprehension, Wo



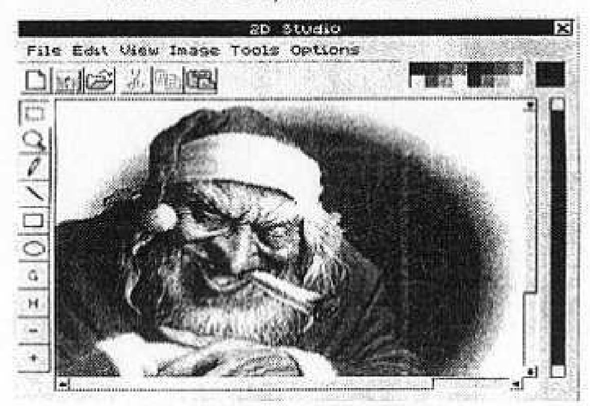
Bisher unveröffentlichte Bilder vom 1. SPC-Treffen am 10. Novemeber 1991
Unreleased pictures from the first SPC meeting on 10th November 1991

ALCHNEWS

AND

Z88 USER

ISSUE 36, MAY 2001



INSIDE THIS ISSUE:

SPRINTER–21ST CENTURY SPECTRUM
THE "LINX PC" : THE Z88'S MISSING BROTHER?
SIMON GOODWIN'S *SERIAL CONNECTIONS* PART 2
+3 SOUND DISTORTION CURE
KEN BEER'S *BACK TO BASICS*

1

AlchNews & Z88 User

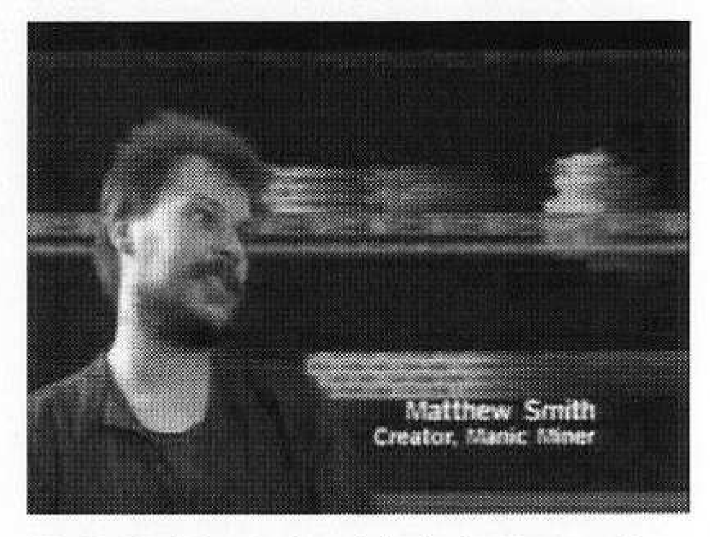
Nachdem ich im letzten Heft die Einstellung von „Desert Island Disk" verkünden musste, nun endlich mal wieder etwas erfeuliches am Spectrum Papermag Himmel.

Andy Davis ist wieder da! Das heisst, er war eigentlich nie verschwunden. Am 27. Juni erreichte mich völlig überraschend ein Mail:

... How are you? It has been such a long time since we last wrote... Did you get the new paper issues of AlchNews? Are you still doing your magazine as I've not seen it for such a long time... Andy Davis

Natürlich habe ich sofort zurückgemailt und wir konnten auch schnell ein Mißverständnis ausräumen. Somit werden wir in Zukunft wieder unsere Magazine und Artikel austauschen.

Inzwischen liegen mir vier Ausgaben vor, angefangen mit Nr. 33. Durchschnittlicher Umfang 32 Seiten, inhaltlich beschäftigt sich das Magazin neben dem Spectrum ebenso wie unseres mit Emulatoren und statt des SAM unterstützt es den beliebten Z88.



Ein Beispiel aus dem Inhalt der letzten Ausgabe Nr. 36. Dort las ich von einem einstündigen Fernsehspezial namens „Thumb Candy" im März auf Channel 4. Inhalt: Die Geschichte der Videospiele. Ein Teil der Sendung hätte sicher auch hierzulande interessiert: Ein Gespräch mit Matthew Smith, der dabei in seinem Schlafzimmer sass und auf einem Gummispectrum sein (!) „Manic Miner" spielte. Obiges Bild davon habe ich von der Webseite

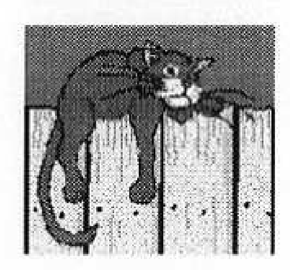
http://www.jonlan.demon.co.uk/ spectrum/matsmith/

Für Interessierte: Das Magazin erscheint alle 2 Monate zum Preis von 1,50 Pfund. Wer es beziehen möchte, schreibe bitte an:

Andy Davis
1 Lichfield Avenue, Astley
Manchester, M29 7WW, England

Roelof Koning:

MegaCat - Teil 2



Nachfolgend der Machinencode für eine Sektor-Lade-Routine. Das ganze kann in einem Assembler (z.B. Freeware 'Tornado') eingetippt werden. **OPUS**-Freunde sollen die Zeilen die mit '%%' anfangen nicht eintippen, und **MGT**-Leute (**DISCiPLE und Plus-D**) sollen die Zeilen mit '$$' überspringen. Weil ich momentan kein Disciple oder Plus-D betriebsfähig habe, konnte ich diese Versionen nicht testen. OPUS arbeitet bei mir hervorragend, habe Probefahrten mit drei unterschiedlichen Laufwerken gemacht (Panasonic 1.44, Sony 1.44, und TEAC 720). Und Disketten von etwa zehn Systemen. Das alles ist garnicht einfach gewesen. Weil es Disketten gibt die schon auf PC als doppelseitig mit einer Sektorlänge von 512 Bytes vorformattiert waren bevor Sie auf OPUS einseitig mit Sektoren von 256 Bytes formattiert wurden. Dann findet man hinten andere Sektoren als vorne, weil die alten noch immer da sind....

Wie man sehen kann ist der Text teils noch 'henglisch', das ist die holländisch-englische Sprache die ich zu Hause beim programmieren verwende. Diese Sprache sollte hauptsächlich Stichworte liefern, damit ich später noch weiss, was vorher damals gemeint war. Ist aber doch möglich, dass es für andere Profis (sind wir doch) auch einiges aufklärt...

Arbeiten soll das Ganze (Opus genau wie beide MGT's) etwa wie:

```
10 REM RESET Laufwerk 1, INQUIRE
   Diskette
12 LET drv=50011: POKE drv,0
15 LET L=USR 50000:IF L<>0 THEN
   PRINT "error":STOP:
20 PRINT "wprot";PEEK 50023'
   "sides";PEEK 50017
21 PRINT "tracks";PEEK 50018':
   "sectors";PEEK 50019
22 PRINT "sectlen";PEEK 50015 +
   256*PEEK 50016
23 PRINT "sectrange";PEEK 50020;
   "-";PEEK 50021'"intlv";PEEK
   50022
26 STOP
20 REM LADE Seite0, Spur6,
   Sektor 5 als Bits auf den
   Bildschirm.
25 LET adr=50009: POKE adr,0:
   POKE adr+1,64: REM =16384
30 LET seite=50012: POKE seite,
   0:LET sect=50013:POKE sect,5:
   LET trk=50014: POKE trk,6
40 LET L=USR 50003:IF L<>0 THEN
   PRINT "error":STOP
```

Man kann durch POKEn anderer Werte an die Adressen 'memaddrs', 'drivenr', 'sidenr', 'sectnr', und 'tracknr' (Bei ORG=50000 sind das 50009/10, 50011, 50012, 50013 und 50014) Sektoren auf einer Diskette untersuchen. Und man kann herausfinden, ob 720K Disketten von anderen Systemen sich auch annähern lassen. Weil dass ja das Ziel ist.

Wenn's soweit geklappt hat, sind wir dran. Jetzt müssen wir sehen, wie man Disketten unterschiedlicher Systeme erkennen kann. So dass wir das Layout verarbeiten können und die User-Data richtig verstehen....

In späterem Machinencode werde ich die Routinen auf ähnliche Weise wie hier in BASIC benutzen, dass heisst zuerst einige Pokes und dann eine von beiden aufrufen. Also die Schnittstellen-abhängige Programmierung ist jetzt vorbei, hoffe ich!

Nächstes Mal mehr, Kommentar willkommen.

Roelof.Koning@12move.nl

```
;———— *TORNADO* ————————————
;Sektor Lade-Routine. Roelof Koning 2001. (SPC)
;Zeilen mit:   $$     sind nur fuer OPUS
;              %%     sind nur fuer Disciple/Plus-D
;              %%!!   = Unterschied Disciple/Plus-D
;Die Kennzeichen '$$', '%%' und '!!' nicht eintippen!
```

```
;------------------------------------------------------------
            ORG  50000  ; herstellen fuer diese Adr.
            DUMP 50000  ; herstellen auf diese Adr.
start1      JP   dsk_inqui
start2      JP   loadsect
start3      DEFB 0,0,0    ;spare entry
;---- MGT only-------------------------------------------
%%!! com_r  EQU 27  ;=Disciple!    227 =Plus-D!
%%!! lat_r  EQU 31  ;                239
%%!! trk_r  EQU 91  ;                235
%%!! sec_r  EQU 155 ;                243
%%!! dat_r  EQU 219 ;                251
;------------------------------------------------------------
memaddr     DEFW 0              ;vorher poken !!!
drivenr     DEFB 0              ; 0 or 1
sidenr      DEFB 0              ; 0 or 1
sectnr      DEFB 0              ; any valid number
tracknr     DEFB 0              ; any valid number
;------------------------------------------------------------
sectlen     DEFW 250            ;to be read !!!
maxside     DEFB 250            ;default filled with
maxtrk      DEFB 250            ;    250 for testing
maxsect     DEFB 250            ;nr of sects/track
minsectnr   DEFB 250            ;lowest sector nr
mxsectnr    DEFB 250            ;highest sect nr
intlv       DEFB 250            ;interleave factor
writprot    DEFB 250            ;"write protected'
dsktype     DEFB 250            ;which system
;------------------------------------------------------------
table1      DEFS 12             ;make some room
table2      DEFS 45             ;  and some more
trkstore1   DEFB 222            ;meaningless nr,
trkstore2   DEFB 222            ;          for testing

;############### START 1 ################
;Diese Routine reseted das Laufwerk: Kopf auf Spur 0.
;Daneben wird noch Info ueber die Diskette gesammelt.
;------------------------------------------------------------
dsk_inqui   DI
$$          CALL 5896           ;Page OpusROM in
            CALL dsk_inq2
di_exit     EI
            LD  B,0             ;return value 'USR'
            LD  C,A
$$          CALL 5960           ;OpusROM out
            RET                 ;Carry is error
;------------------------------------------------------------
;0= disk present,  1= nodisk,  128= drive not ready
;208= Break,        4-28= disk error
;------------------ disk present? --------------------
dsk_inq2    CALL setdrvsd       ;prep relevant bits
%%          IN   A,(com_r)      ;status register MGT
$$          LD   A,(#2800)      ;status FDC OPUS
            AND  %10000000      ;test Motor On bit
            SCF                 ;prepare error flag
            RET  NZ             ;drv not ready
            LD   A,%11010000    ;force int command
            CALL ldcomreg       ;give command
            LD   HL,0           ;reset time out cnter
            LD   C,2            ;mask IndexPulse
            CALL  testIP        ;test 'on'
```

```
            RET  C              ;time out, no disk
            LD   C,0            ;mask IndexPulse
            CALL testIP         ;test 'off'
            RET  C              ;time out, no disk
;-------------- test tracks & reset drive --------------
            CALL dsk_rest       ;restore drive
            RET  C              ;on error
            LD   A,79           ;last track
%%          OUT  (dat_r),A      ;data reg MGT
$$          LD   (#2803),A      ;data register OPUS
            LD   A,%00010100    ;signal 'seek'
            CALL doread2        ;command the FDC
            LD   A,80           ;signal nrs of tracks
            JR   NC,res2        ;no error
            LD   A,255          ;some value, error
res2        LD   (maxtrk),A
            CALL dsk_rest       ;final drive reset
            RET  C              ;on error
;-------------- test side 2 ------------------------------
            LD   A,1
            LD   (sidenr),A     ;signal 'side 2'
            LD   HL,table1
            LD   (memaddr),HL   ;load data here
            LD   HL,table2      ;store sectnrs here
            CALL rdaddrs        ;read sector address
            LD   A,0            ;single side if error
            JR   C,pokeside     ;side 2 not found
            LD   A,(table1 +1)  ;get side nr found
pokeside    INC  A              ;adjust to 1 or 2
            LD   (maxside),A
;-------------- inq nr of sects --------------------------
            XOR  A              ;signal 'side 1'
            LD   (sidenr),A
            LD   HL,table2      ;store sectnrs here
            CALL rdaddrs        ;get a sector nr
            RET  C              ;errorflags in A reg
            LD   C,A            ;copy sect nr found
            LD   B,0            ;counter
inqtr1      PUSH BC
            CALL rdaddrs        ;get sector number
            POP  BC
            RET  C              ;error, flags in A reg
            INC  B
            CP   C              ;first sector again?
            JR   NZ,inqtr1      ;nz= a new sectornr
            LD   A,B            ;counter
            LD   (maxsect),A    ;nr of sectors
;-------------- calc sect len ----------------------------
            LD   A,(table1 +3)  ;sector length code
            INC  A              ;adjust
            LD   B,A
            LD   HL,64
calcl       ADD  HL,HL          ;double 'B' times
            DJNZ calcl
            LD   (sectlen),HL   ;128,256,512,1024
;-------------- find min/max sect nrs --------------------
            LD   A,(maxsect)    ;number of sectors
            LD   B,A            ;in counter
            LD   HL,table2
            LD   E,(HL)
            LD   D,E            ;get first entry in tbl
```

```
secnr2      INC HL
            LD  A,(HL)          ;get next entry
            CP  E
            JR  NC,secnr3       ;find smallest nr
            LD  E,A             ;keep smallest
secnr3      CP  D
            JR  C,secnr4        ;find highest nr
            LD  D,A             ;keep highest
secnr4      DJNZ secnr2
            LD  (minsectnr),DE  ;store sectnr range
;-------------- calc interleave factor --------------
            LD  HL,table2       ;sectnrs stored here
            PUSH HL
            POP  DE             ;DE = HL
            LD  A,(HL)          ;first sectornr in tbl
            LD  C,A
            INC C               ;C holds 'next' nr
cintlv2     INC HL              ;step through table
            LD  A,(HL)
            CP  C               ;find 'next number'
            JR  NZ,cintlv2
            SBC HL,DE           ;calc pos. 'next nr'
            LD  A,L             ;this is interleave!
            LD  (intlv),A
            XOR A               ;no error
            RET
```

```
;############### START 2 ##################
;Diese Routine laedt einen Sektor in den Speicher. Zuerst
;werden 'drivenr' und 'sidenr' festgelegt. Wenn dann der
;Lade-Befehl gegeben wurde, werden die 'fertigstehenden'
;Bytes abgeholt.
;NB bei Opus generiert der FDC fuer jedes zu lesende Byte
;einen Non Maskable Interrupt. Diese NMI's werden verar-
;beitet, waehrend die Routine ueberprueft ob der FDC fertig
;ist. Disciple/Plus-D warten auf 'normale' Weise auf das
;DRQ Signal.
;------------------------------------------------------
loadsect    DI
$$          CALL 5896           ;page OpusROM in
            CALL loadsect2
            JR  C,err_out       ;error
            EX  AF,AF'          ;save value in A
%%          PUSH HL             ;last memaddr + 1
            CALL savetrk        ;copy reg to trkstore
%%          POP  DE             ;next memaddr
            EX  AF,AF'
err_out     LD  B,0
            LD  C,A             ;return value 'USR'
$$          CALL 5960           ;Opus ROM out
            EI
            RET                 ;carry = error
;------------------------------------------------------
loadsect2   CALL setdrvsd       ;prep relevant bits
            LD  HL,(memaddr)    ;destination in mem
            LD  DE,(sectnr)     ;e= sector, d= track
            PUSH HL             ;memory address
            CALL gettrk         ;restore track reg
            CP  D               ;arm on this track?
            JR  Z,seekdon
            LD  A,D             ;track number
%%          OUT (dat_r),A       ;new trk in datareg
$$          LD  (#2803),A       ;new trk in datareg
            LD  A,%00010000     ;signal 'seek track'
            CALL ldcomreg       ;execute seek
seekdon     POP HL              ;memory address
            RET C               ;on error
            LD  A,E             ;wanted sector
%%          OUT (sec_r),A       ;in sector reg MGT
$$          LD  (#2802),A       ;in sector reg OPUS
$$          EX  DE,HL           ;DE = mem address
            LD  A,%10000000     ;signal 'load sector'
doread      EQU $
$$          LD  HL,nmiread      ;point to read routine
doread2     CALL ldcomreg       ;excute load sector
            RET C               ;exit on error
;----- Ladeschleife MGT ---------------------------
%%          LD  BC,dat_r        ;address of Data reg
%%          JR  tstdrq          ;jump into load loop
%%ldlop      INI                 ;rd port 'C' into (HL)
%%tstdrq     IN A,(com_r)        ;read status register
%%          BIT 1,A             ;test 'byte present'
%%          JR  NZ,ldlop        ;if so, rd and store
%%          BIT 0,A             ;test for FDC ready
%%          JR  NZ,tstdrq       ;loop if still busy
;------------------------------------------------------
            AND %00011100       ;keep errorbits
            RET Z               ;exit now if OK
            SCF                 ;signal error in carry
            RET
;------ OPUS only ---------------------------------
;Diese Routine liest bei jedem NMI ein Byte vom FDC. Auf
;NMI-Adresse #66 steht JP(HL), und HL enthalt die
;Adresse dieser Routine.
;------------------------------------------------------
$$nmiread   EX  AF,AF'
$$          LD  A,(#2803)       ;get byte from FDC
$$          LD  (DE),A          ;move into memory
$$          INC DE              ;next address
$$          EX  AF,AF'
$$          RETN                ;return from interrupt
;------------------------------------------------------
```

```
;################### SUB Routines ##########
;Drive und Side werden aufgrund von Bits gewaehlt, die in
;Port #1F (Disciple) oder #EF (Plus-D) festgelegt sind.
;'SideSelect' und 'DrvSelect' werden bei Disciple anders
;festgelegt wie bei Plus-D! Bei Opus werden diese Bits in
;ein PIA 6821 Port festgelegt. Diese Port ist 'memory
;mapped', auf Speicheradresse #3000.
;------------------------------------------------------
setdrvsd    LD  D,0             ;prepare pattern in D
            LD  A,(sidenr)
            AND A               ;test side, 0-1
            JR  Z,sds1          ;side 0
%%          SET 1,D             ;side 1, DisCiple
%%!!        SET 7,D             ;        Plus-D
$$          SET 4,D             ;        OPUS
sds1        LD  A,(drivenr)     ;(0-1)
            INC A               ;adjust
            CP  1               ;drive 1
            JR  Z,sds2          ;jp if not drv2
```

```
%%          LD  A,0              ;Disple  000000xx
%%!!        LD  A,2              ;Plus-D x00000xx
$$          LD  A,2              ;OPUS  000x00xx
sds2        OR  D                ;pattern in D
            LD  D,A              ;save briefly
            CALL fdc_ready       ;change when free
            LD  A,D
%%          OUT (lat_r),A        ;write to latch
$$          LD  A,(#3000)        ;read 6821 pia port
$$          AND %11101100        ;drop drv/side bits
$$          OR  D                ;add mask
$$          LD  (#3000),A        ;write to pia port
            RET
;-----------------------------------------------------
savetrk     CALL setstore
%%          IN  A,(trk_r)        ;read track register
$$          LD  A,(#2801)        ;read track register
            LD  (HL),A
            RET
;-----------------------------------------------------
gettrk      CALL setstore
            LD  A,(HL)
%%          OUT (trk_r),A        ;write track register
$$          LD  (#2801),A        ;write track register
            RET
;-----------------------------------------------------
setstore    LD  HL,trkstore1     ;table start
            LD  A,(drivenr)
            AND A                ;test for '0'
            RET Z                ;exit for drive 1
            INC HL               ;point to trkstore2
            RET
;— restore -------------------------------------------
;Reset-Routine fuer Drive: Kopf auf Spur 0. Der FDC
;(Floppy-Disc-Controller) lagert die SpurNummer in das
;'track register', daneben wird es auch noch in 'trkstore'
;abgespeichert. Wenn zwei Laufwerke benutzt werden,
;soll immer die 'andere' Spur-Nummer aufbewahrt werden.
;-----------------------------------------------------
dsk_rest    LD  A,%00001000      ;no spinup, no verif
            CALL ldcomreg        ;exec 'reset drive'
            RET C                ;BREAK pressed!
            LD  C,A              ;store status briefly
            AND %01000000        ;keep writprot flag
            LD  (writprot),A
            LD  A,C              ;status
            AND %00011000        ;keep errorbits
            SCF                  ;prepare error flag
            RET NZ               ;exit error, A=8-24
            JP  savetrk          ;read trk reg + store
;--------test index pulse ----------------------------
testIP      EQU $
%%          IN  A,(com_r)        ;status reg
$$          LD  A,(#2800)        ;status FDC
            XOR C                ;IndexPulse mask
            AND 2
            RET Z                ;ret if disk in drive
            LD  B,2
w2          DJNZ w2              ;some extra delay
            DEC HL
            LD  A,H              ;time out counter
```

```
            OR  L
            JR  NZ,testIP        ;total 65535 times
            INC A                ;A=1
            SCF                  ;signal error
            RET                  ;time out, no disk
;-----------------------------------------------------
;Diese Routine liest SpurNr, SeiteNr, SektorNr, usw.
;vom ersten Sektor, der am Lese-Kopf vorbeikomt.
;-----------------------------------------------------
rdaddrs     PUSH HL              ;save table addrs
            CALL setdrvsd        ;prepare these bits
            LD  DE,(memaddr)     ;dest in mem
            LD  A,%11000000      ; 'read address'
            CALL doread          ;read 6 bytes
            POP HL               ;table for sector nrs
            RET C                ; error
            LD  A,(table1 +2)    ;get sect nr found
            LD  (HL),A
            INC HL               ;move into table
            RET
;-----------------------------------------------------
;Gebe ein Kommando an FloppyDiskController
;-----------------------------------------------------
ldcomreg    EX  AF,AF'           ;save command
            CALL FDCbusy         ;is FDC free?
            EX  AF,AF'
%%          OUT (com_r),A        ;into command reg
$$          LD  (#2800),A        ;into command reg
            LD  B,8              ;wait for 32 usec
wait        DJNZ wait            ;before rd the status
            JR  no_spac          ;test for FDC 'ready'
;-----------------------------------------------------
;Hier wird geprueft ob der FloppyDiskController fertig
;steht. Der FDC koennte ja beim letzten Auftrag stehen
;geblieben sein. Als Notausgang fungiert 'Break'.
;-----------------------------------------------------
FDCbusy     LD  A,127            ;test for 'BREAK'
            IN  A,(254)
            RRA                  ;'SPACE' key?
            JR  C,no_spac
            LD  A,254
            IN  A,(254)
            RRA                  ;'CAPS' key?
            JR  NC,force_int
no_spac     EQU $
%%          IN  A,(com_r)        ;status reg MGT
$$          LD  A,(#2800)        ;status reg OPUS
            BIT 0,A              ;FDC ready?
            JR  NZ,FDCbusy       ;loop if not
            AND A                ;reset carry flag
            RET                  ;no error
;-----------------------------------------------------
;Bei "Break' wird das FDC-Kommando in 'A' in einen
;'Notbremse-Befehl' geaendert. Bei Rueckkehr in die
;Haupt-Routine wird diese letzte wegen des Carryflag sofort
;abgebrochen.
;-----------------------------------------------------
force_int   LD  A,%11010000      ;'forced interrupt'
            SCF                  ;set error flag
            RET                  ;exec command
;end -------------------------------------------------
```



ASCD 0.90 WIP 1

(july 2001)

This time a new and very interesting SAM emulator is worth to be mentioned. It´s from Aley Keprt and in my eyes the unique is, that it also emulates ZX Spectrum programs, 48K and 128K. Here is a brief description with added comments by me (Wo).

ASCD = "Aley's modified SimCoupe for DOS" - a Sam Coupe & ZX Spectrum 48/128 emulator

(works also under Windows 98 in DOS mode)

Copyright (c) Aley Keprt 1998-2001, based on other programs (see below)
Distributed under GNU GPL license.

required: PC AT 486DX4/120MHz, VGA, MS-DOS or Windows 95/98/ME/NT/2000
recommended: PC AT Pentium 100MHz, PCI VGA, SoundBlaster 16, DOS or Windows or something
important keys: F2=Menu, Ctrl+F12=Exit

Important!

1. If you don't know what is Sam Coupé and/ or ZX Spectrum, please read some information first. This is documentation for the emulator, not for emulated computers. See for example:
http://get.to/samcoupe
http://www.void.jump.org

(Anyone here who doesn´t know what a SAM or Spectrum is?)

2. If you don't have ROM images, please download file "ascd_sup.zip" from the location where you downloaded the emulator. It contains these files:
spectrum.rom, zx128_0.rom, zx128_1.rom, sam_rom0.rom, sam_rom1.rom, and also cwsdpmi.exe (a file needed to run emulator in plain MS-DOS without Windows).

ASCD features:

* Video emulation engine which simulates clut and vmpr switching on line interrupts
* Timing of Z80 instructions for accurate CPU emulation
* Nice text-mode user interface for disk selection, etc.
* Quality SAA1099 and AY8910 audio emulation
* ZXS/Sam beeper emulation (line-based, i.e. accuracy is 15.6kHz)
* Sam and ZX Spectrum Keyboard emulation modes, with mapping to PC keyboard
* Supports all kinds of real PC joysticks, and joystick emulation on keyboard
* All ROM images are legally distributed with the emulator
* GZ-compressed file are supported (any file can be .gz packed)
* Can be compiled with the latest GNU gcc (currently djgpp 2.03 and compatible)

Note: The original SimCoupe can't be compiled today, since it relies on some hidden bugs of the previous versions of gcc, which have already been fixed.

Building ASCD

Get the latest djgpp, and compile it all together.
You will need also SAAemu, Mame, and zlib.
Good luck! You will need it :-)

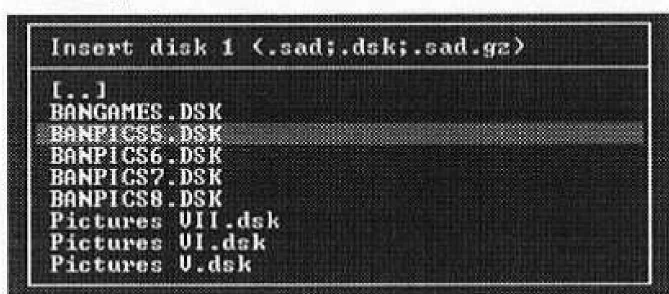
Running ASCD

Please always download UP-TO-DATE version of "ascd_sup.zip" which contains some required files for emulation (ROM images and DPMI server). It is distributed separatetly to conserve size of ASCD itself.

If you want to emulate the ZX Spectrum instead of the SAM Coupe then use "ascd -zx" or "ascd -128". You can also switch to ZXS emulation mode by loading a snapshot from menu (press F2 to enter menu).



(Loading a snapshot from the filemenu is the far better way - and the best is, it works!)

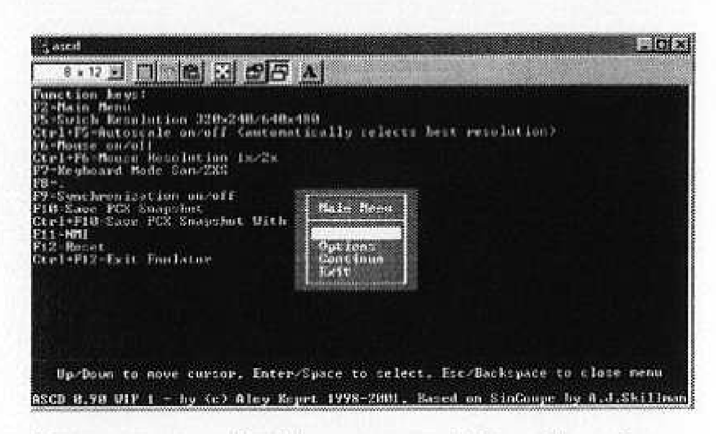


(However - one thing could be changed: to hold the folder where you last was. Here you have to go through all the folders again.)

ASCD can be also run from the DOS box in Windows. In this case Windows provides the DPMI, so CWSDPMI.EXE is not required.

Using ASCD

Most of the Sam keyboard has been mapped to a PC style keyboard, eg. using the PC keys for symbols will produce the correct response on the Sam. Certain Sam specific keys are mapped onto special keys on the PC keyboard:



(As you - probably - see, all function keys are shown in this menu)

Symbol:	Left Ctrl or Right Ctrl
Edit:	RightAlt key
F0 - F9:	Numeric keypad 0-9 (possibly with Num Lock)
Inv:	Keypad Enter
(c):	Keypad .

The emulator is controlled using the PC function keys F1-F12. Press F2 to enter menu and you will see list of all supported F-key commands (on-screen help). And also don't forget: Ctrl+F12 = Exit to DOS.

Note: As was pointed out above ASCD has two keyboard modes, Sam and Spectrum. The former is default at startup. When using SimCoupe to emulate a Spectrum, the latter is default. The Spectrum keyboard mode may be incomplete, but the main functions are there :)

The F6 key controls the Sam Mouse Mode, by default the emulator acts as if a Sam mouse is not connected to the machine.

(You can use your PC mouse like a SAM mouse, I have tried it out!)

Tapping F6 once turns the mouse emulation on. In this mode the mouse is in low sensitivity mode - registering 256 pixels in the x direction. If you use an application which expects full 512 pixel mouse sensitivity, press Ctrl+F6 to enable high (double) resolution.

The F5 key switches between fast low resolution 320x200 graphics, and slower high resolution 640x480 graphics. Although the latter one is slower, it is much better, since it is able to display mode 3 (512x192) graphics, and has correct aspect ratio.

(As seen below.)

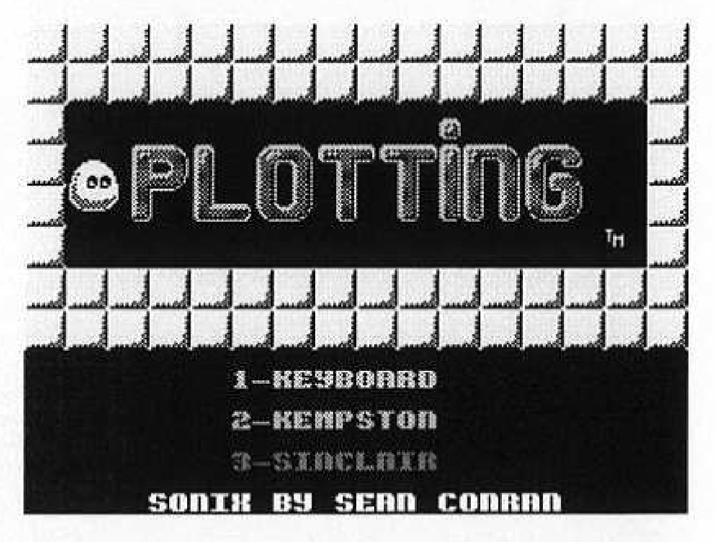


Disk Usage
ASCD supports:
* DSK/Sam images (800KB raw data)
* DSK/MS-DOS images (720KB raw data)
* SAD v1 images (Aley's Sam Backup format)
* SAD v2 images (Aley's Sam Backup format, packed with GZip)

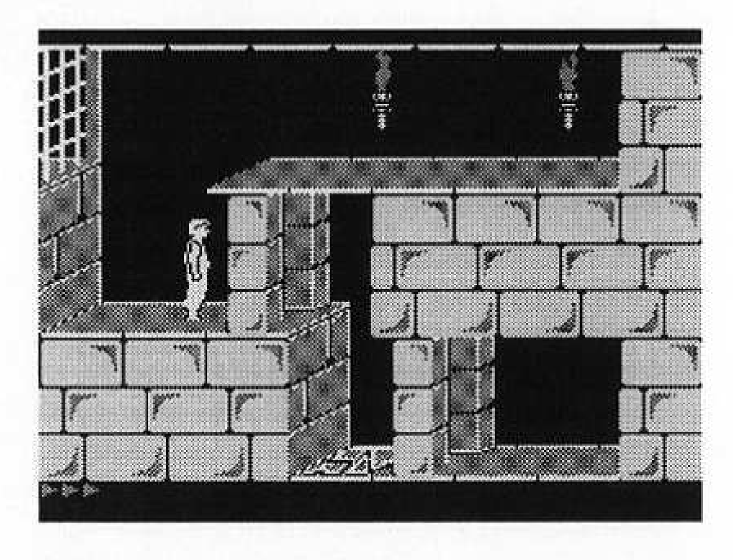
You can use 2sad utility (available for Win32 and DOS) to convert all existing images to packed SAD v2 and save much disk space!

ASCD Options
Please type "ascd -help" to see the complete list of available command line options.

ZX Spectrum Emulation
You can simply start ZX Spectrum emulation by loading a snapshot. Supported are two most standard file types in all (four) versions: .sna & .z80. You can also load .z80.gz files compressed by gzip.



(This two-in-one solution, SAM and Spectrum included 128K support, is a great improvement and works very well. If you load a .Z80 file you automatically are in the right Spectrum mode. But I miss a little bit the .tap file)



AY8910 sound emulation is always on, even when woriking in 48k mode. I hope this was clever decision.
Tape is not supported at all. See command line options for more information.

Input Recording
ASCD supports also Input Recording by utilising AIR library. This functionality is supposed mainly for game tournaments, so it can't be distributed in source form. Since this is against GPL license, ASCD with AIR library is not publicly distributed at all, and if

you have public version of ASCD, then you surely miss AIR functionality in there.

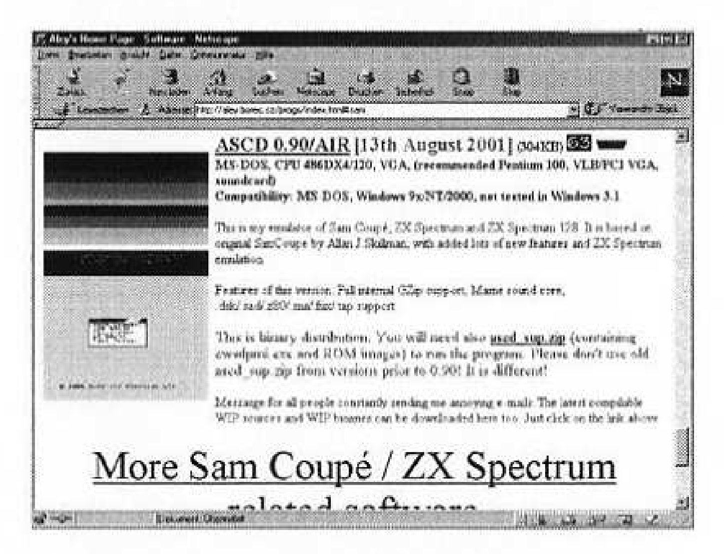
Acknowledgments

ASCD was designed and programmed by Aley Keprt, copyright (c) 1998-2001.
It is partially based on original SimCoupe, from which it has been rewritten.

It also consists of SAAemu, zlib, and very little portion of MAME.

http://get.to/samcoupe
http://www.zlib.org
http://www.mame.net

Special tournament version contains also AIR library - by Aley Keprt 2001.



„Last minute message" from Aley:

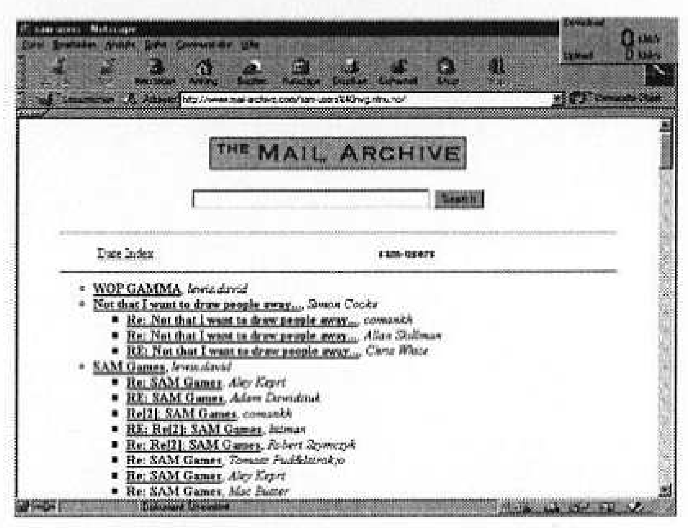
Since *some* people still sending me annoying messages, I simple must release a new version including source code.

Due to new download counters, no really direct download is supported. Just go to

http://aley.borec.cz

and click on ASCD 0.90. Then you can download all supported new and old versions. Enjoy!

SAM Mail Archive



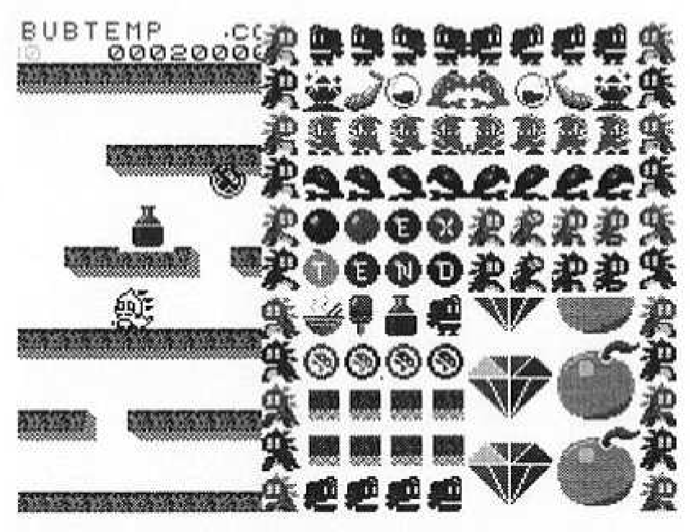
A lot of us SAM friends are mailing via the sam usergroup ***sam-users@nvg.ntnu.no***. This can easily fill up your mailfolder and maybe you will find yourself within dozends of unsorted mails. Don´t worry, help is on it´s way. On Monday, 20th August 2001 Frode Tenneboe surprised us with this message:

Hi! Have just created a mail archive to archive the contents from this list. You'll find it at:

http://www.mail-archive.com/sam-users%40nvg.ntnu.no/

So let them post keep coming!

Right on, take a look at this. *Wo*



This screenshot shows, how „sprites" are developed (and fills this space here).



Jeder von uns kennt die langsamen Fill-routinen von "The Hobbit". Eine Fill-Routine kann eine beliebige Figur am Bildschirm mit einer Farbe füllen. Weil jede Figur anders aussieht könnte man denken, dass eine solche Routine für alle Figuren schwer zu schreiben sein sollte.
Eine Fill-Routine ist eigentlich ganz einfach! Wenn diese rekursiv programmiert ist! Im BASIC sieht eine Fill-Routine etwa so aus:

```
  10 REM fill basic
  20 CIRCLE 100,100,10
  30 LET x=100: LET y=100: GO
     SUB 1000
  40 STOP
1000 REM Fill-recursief
1010 IF POINT (x,y) THEN RETURN
1020 PLOT x,y
1030 IF y>0 THEN LET y=y-1: GO
     SUB 1000: LET y=y+1
1040 IF y<255 THEN LET y=y+1: GO
     SUB 1000: LET y=y-1
1050 IF x>0 THEN LET x=x-1: GO
     SUB 1000: LET x=x+1
1060 IF x<255 THEN LET x=x+1: GO
     SUB 1000: LET x=x-1
1070 RETURN
```

Dieses Beispiel wird gut gefilled, ist aber etwas langsam! Wenn mann z.B.

```
  20 CIRCLE 100,100,40
```

eintippt, kann man zwei Tassen Kaffee trinken, während der Speccie filled.

Im Maschinensprache wird schneller gearbeitet, und dieselbe Routine sieht dann so aus.

```
          ORG 50000
          DUMP 50000

          LD B,100       ; y-coordinaat
          LD C,100       ; x-coordinaat
          LD A,175
          SUB B
          LD B,A         ; #22B1 wil 175-b

          CALL fill      ; fill (x,y)
          RET

fill      CALL point     ; point-adress
          RET NZ         ; if point(x,y) then
                           return
          OR (HL)
          LD (HL),A      ; plot x,y
          LD A,B
          DEC B
          OR A
          CALL NZ,fill   ; if y>0 then fill
                           (x,y-1)
          INC B
          LD A,B
          INC B
          CP 175
          CALL NZ,fill   ; if y<175 then
                           fill(x,y+1)
          DEC B
          INC C
          CALL NZ,fill   ; if x<255 then
                           fill(x+1,y)

          DEC C
          LD A,C
          DEC C
          OR A
          CALL NZ,fill   ; if x>0 then fill
                           (x-1,y)
          INC C
          RET

point     PUSH DE
          LD A,B
          CALL #22B1
          LD E,A
          INC E
          LD A,(HL)
          LD D,1
pointlus  RLCA
          RRC D
          DEC E
          JR NZ,pointlus
```

```
        AND 1        ; z-flag = set if
                     point= on
        LD A,D       ; d = value for bit-
                     set
        POP DE
        RET
```

In MC nutzt man die Routinen zur POINT-adresse auf #22B1. Gleichzeitig wird auch das richtige Bit zum SET gerechnet (=PLOT). Das gleiche Programm

```
20 CIRCLE 100,100,40:
   RANDOMIZE USR 5E4
```

geht schneller als BASIC mit CIRCLE 100,100,10. Das Programm

```
20 CLS : RANDOMIZE USR 5E4
```

wird aber 'crashen'. Warum? Jedes Bit wird von einem recursiven CALL gefilled. Deshalb bekommt man einen Stack von 176*256*2 = 90112 Bytes!!

Unser Speccie hat nicht soviel Speicher. Der Stack muss daher kleiner gehalten werden. Mit folgender MC-Routine wird nicht jeder Punkt, sondern nur jede Bild-Zeile zusammen mit 2 Bytes der x-Koordinaten zwischen gespeichert, wenn Fill aktiv ist.

```
        ORG 50000
        DUMP 50000

        LD B,100     ; y-coordinaat
        LD C,100     ; x-coordinaat
        LD A,175
        SUB B
        LD B,A       ; #22B1 wil 175-b
        CALL fill
        RET

fill    CALL point   ; point-adress
        RET NZ       ; if point(x,y) then
                     return
        OR (HL)
        LD (HL),A    ; plot x,y
        PUSH BC
beginx  LD A,C
        DEC C        ; x=x-1
        OR A
        JR Z,eindx   ; if x <> 0 then
        CALL point
        JR NZ,eindx  ; if not point
                     (x-1,y) then
        OR (HL)
        LD (HL),A    ; plot x-1,y
        JR beginx
eindx   INC C        ; 1x zuviel -1, also
                     +1
        LD E,C       ; e=bx
        POP BC

ex2     INC C        ; x=x+1
        JR Z,ex3     ; if x <> 255 then
        CALL point   ; if not point(x+1,y)
                     then
        JR NZ,ex3
        OR (HL)
        LD (HL),A    ; plot x+1,y
        JR ex2
ex3     LD D,C       ; d=ex+1
while   LD C,E       ; Anfang lus
        LD A,B
        OR A
        JR Z,oben    ; y=0, nur nach
                     oben
        PUSH DE      ; Stack x-coor-
                     dinaten
        DEC B
        CALL fill    ; fill (x,y-1)
        INC B
        POP DE
        LD C,E
oben    LD A,B
        CP 175
        JR Z,ince    ; y=175, nicht
                     nach unten.
        PUSH DE
        INC B
        CALL fill    ; fill (x,y+1)
        DEC B
        POP DE
ince    INC E
        LD A,E
```

```
         CP D
         JR NZ,while
         RET

point    PUSH DE
         LD A,B
         CALL #22B1
         LD E,A
         INC E
         LD A,(HL)
         LD D,1
pointlus RLCA
         RRC D
         DEC E
         JR NZ,pointlus
         AND 1
         LD A,D
         POP DE
         RET
```

Diese Routine wird für 99% aller Fill's gut sein. Nur wenn eine Figur sehr oft wie eine Schlange am Bildschirm von oben nach unten geht, wird diese Routine einen grossen Stack bekommen und 'crashen'.

Es geht aber noch schneller. Jetzt wird jedes horizontale Bit getestet. Man kann aber auch ein ganzes Byte testen. Statt ein Bit in (HL) kann man auch (HL) auf 0 testen und dann sofort 8 bits 'setten'. Die Routine wird dann sein:

```
         ORG 50000
         DUMP 50000

         LD B,100      ; y-coordinaat
         LD C,100      ; x-coordinaat
         LD A,175
         SUB B
         LD B,A        ; #22B1 wil 175-b
         CALL fill
         RET

fill     CALL point    ; point-adress
         RET NZ        ; if point(x,y) then
                       return
         OR (HL)
         LD (HL),A     ; plot x,y
         PUSH BC
beginx   LD A,C
         DEC C         ; x=x-1
         OR A
         JR Z,eindx    ; if x <> 0 then
         CALL point
         JR NZ,eindx   ; if not point(x-1,y)
                       then
         LD D,(HL)     ; d = alte Wert (HL)
         OR (HL)
         LD (HL),A     ; plot x-1,y
         LD A,D
         OR A
         JR NZ,beginx
         LD (HL),255   ; gleich 8 Bits
                       'setten'
         LD A,C
         AND 248
         LD C,A        ; x-coordinate
                       wird 7 weniger
         JR beginx
eindx    INC C         ; 1x teveel ver-
                       laagd, dus +1
         LD E,C        ; e=bx
         POP BC
ex2      INC C         ; x=x+1
         JR Z,ex3      ; if x <> 255 then
         CALL point    ; if not point(x+1,y)
                       then
         JR NZ,ex3

         LD D,(HL)     ; d = alte Wert (HL)
         OR (HL)
         LD (HL),A     ; plot x+1,y
         LD A,D
         OR A
         JR NZ,ex2
         LD (HL),255   ; wieder 8 Bits
                       'setten'
         LD A,C
         OR 7
         LD C,A        ; x := x + 7
         JR ex2
ex3      LD D,C        ; d=ex+1
while    LD C,E        ; begin van de lus
```

```
        LD A,B
        OR A
        JR  Z,oben
        PUSH DE
        DEC B
        CALL fill
        INC B
        POP DE
        LD C,E
oben    LD A,B
        CP 175
        JR Z,ince
        PUSH DE
        INC B
        CALL fill
        DEC B
        POP DE
ince    INC E
        LD A,E
        CP D
        JR NZ,while
        RET

point   PUSH DE
        LD A,B
        CALL #22B1
        LD E,A
        INC E
        LD A,(HL)
        LD D,1
pointlus RLCA
        RRC D
        DEC E
        JR NZ,pointlus
        AND 1
        LD A,D
        POP DE
        RET
```

Jetzt wird nur mit alten Attributen gefilled. Mit dem letzten Programm kann man aber auch die INK-Farbe mitgeben. Auch werden die Koordinaten jetzt in den Aufruf mitgenommen. Der Aufruf wird dann

```
PLOT INK USR 5E4+2;100,110 : REM
2 = farbe, 100 = x, 110 = y
```

```
        ORG 50000
        DUMP 50000

        LD HL,(#5C5D)
        PUSH HL
        RST #20
        CALL #24FB
        CALL #2314
        LD (23695),A ; temp. Attributes
        RST #20
        CALL #24FB
        RST #20
        CALL #24FB
        CALL #2314
        LD B,A
        PUSH BC
        CALL #2314
        POP BC
        LD C,A
        POP HL
        LD (#5C5D),HL
        LD A,175
        SUB B
        LD B,A          ; #22B1 wil 175-b
        CALL fill
        LD BC,0         ; BC=0, ende USR
                        = 0
        RET

fill    CALL point      ; point-adress
        RET NZ          ; if point(x,y) then
                        return
        OR (HL)
        LD (HL),A       ; in weze: plot x,y
        CALL attrib
        PUSH BC
beginx  LD A,C
        DEC C           ; x=x-1
        OR A
        JR Z,eindx      ; if x <> 0 then
        CALL point
        JR NZ,eindx     ; if not point(x-1,y)
                        then
        LD D,(HL)
        OR (HL)
        LD (HL),A
        CALL attrib
```

```
          LD A,D
          OR A
          JR NZ,beginx
          LD (HL),255
          LD A,C
          AND 248
          LD C,A
          JR beginx
eindx     INC C         ; 1x teveel ver-
                        laagd, dus +1
          LD E,C        ; e=bx
          POP BC
          PUSH DE
ex2       INC C         ; x=x+1
          JR Z,ex3      ; if x <> 255 then
          CALL point    ; if not point(x+1,y)
                        then
          JR NZ,ex3
          LD D,(HL)
          OR (HL)
          LD (HL),A     ; plot x+1,y
          CALL attrib
          LD A,D
          OR A
          JR NZ,ex2
          LD (HL),255
          LD A,C
          OR 7
          LD C,A
          JR ex2
ex3       POP DE
          LD D,C        ; d=ex+1
while     LD C,E        ; begin van de lus
          LD A,B
          OR A
          JR Z,boven
          PUSH DE
          DEC B
          CALL fill
          INC B
          POP DE
          LD C,E
boven     LD A,B
          CP 175
          JR Z,ince
          PUSH DE
          INC B
          CALL fill
          DEC B
          POP DE
ince      INC E
          LD A,E
          CP D
          JR NZ,while
          RET

point     PUSH DE
          LD A,B
          CALL #22B1
          LD E,A
          INC E
          LD A,(HL)
          LD D,1
pointlus  RLCA
          RRC D
          DEC E
          JR NZ,pointlus
          AND 1
          LD A,D
          POP DE
          RET

attrib    LD A,B
          RLCA
          RLCA
          AND 3
          XOR 88
          LD D,A
          LD E,L
          LD A,(23695)
          EX DE,HL
          AND 7
          OR (HL)
          LD (HL),A
          EX DE,HL
          RET
```

Eigentlich darf man nur 1x pro Charakter Attribute setzen. Da es nicht einfach möglich ist, dies zu bauen, wird daher etwa 8x pro Charakter-Attribut gesetzt. Die Fill-Routine ist aber doch schneller als die Hobbit-Routinen.

Dr BEEP „Johan Koelman"

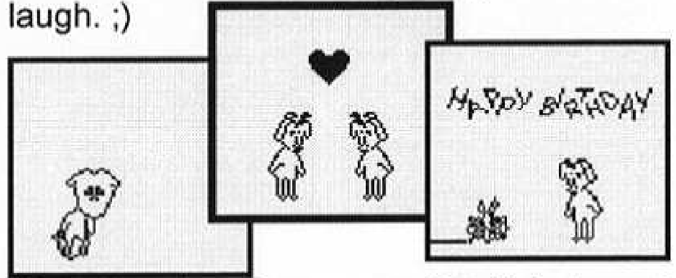
YERZMYEYS "SPREADING-SERVICE"



Hi!

Rabbit demo serie

An interesting stuff is a serie of demos for ZX. It's RABBIT serie I found on GASMAN's *DEMOTOPIA*.
The demos are so fucking funny. Watch and laugh. ;)



(Screens taken from several Rabbit-demos)

New HOOY coder...

And one more news... It seems that HOOY-PROGRAM has a new member, a coder: TIF/H-PRG. Well, welcome aboard. ;)

CC2001 zx-party preresults... I think...

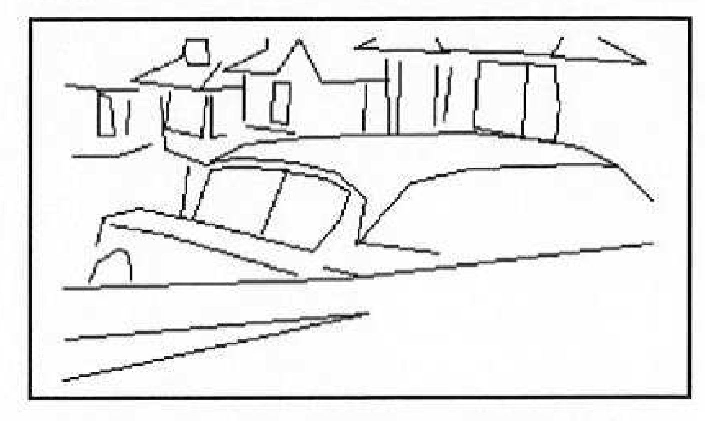
512 byte intro: 1. BoomBoom - Red Alex/CTL; 2.Insult - Asman/Proxium; 3. Nop - Mad Cat

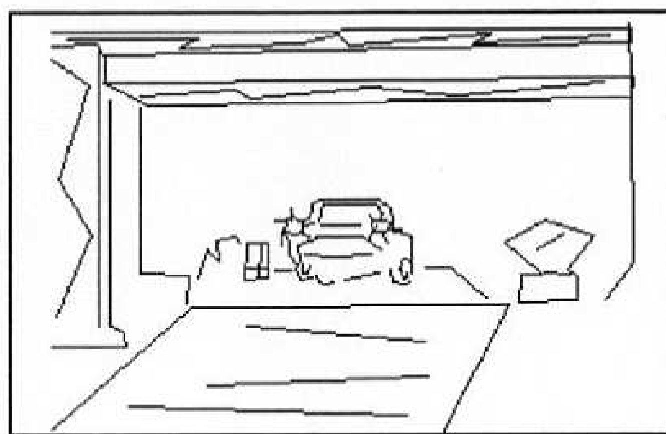
graphics compo: 1. ???? - Pheel/PHT; 2. X-zema - Paracels/Placebo; 3. Among Us - GAS-13

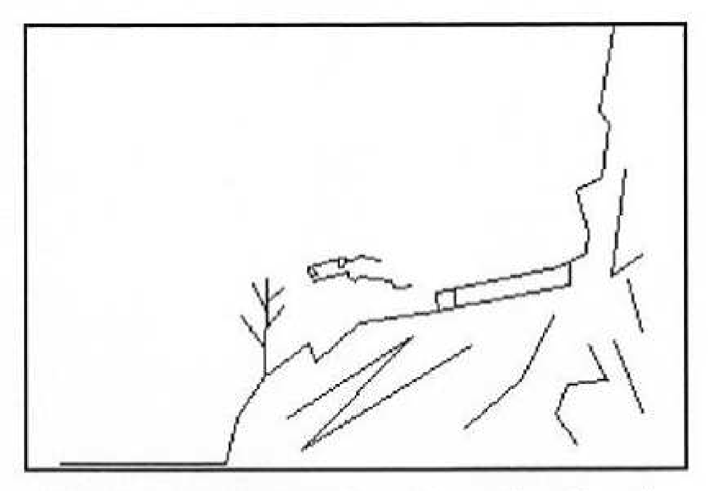
demo compo: 1. Stellar Contour - Brainwawe; 2. Lifeforms - Placebo; 3. Love Gun - VolgaSoft; 4. Jaundice - Skreamer; 5. ???? - Eye-Q

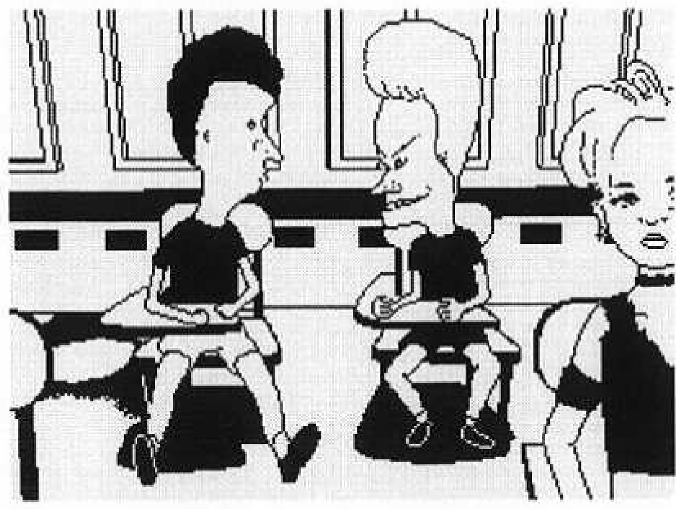
Thanx to Icabod - the owner of the best ZX news site (www.raww.org) I can provide You stuff from CC2001.
Well, BRAINWAVE group once again shown, that they are the best demomakers on Speccy nowadays.

Cu, Yerz/H-PRG





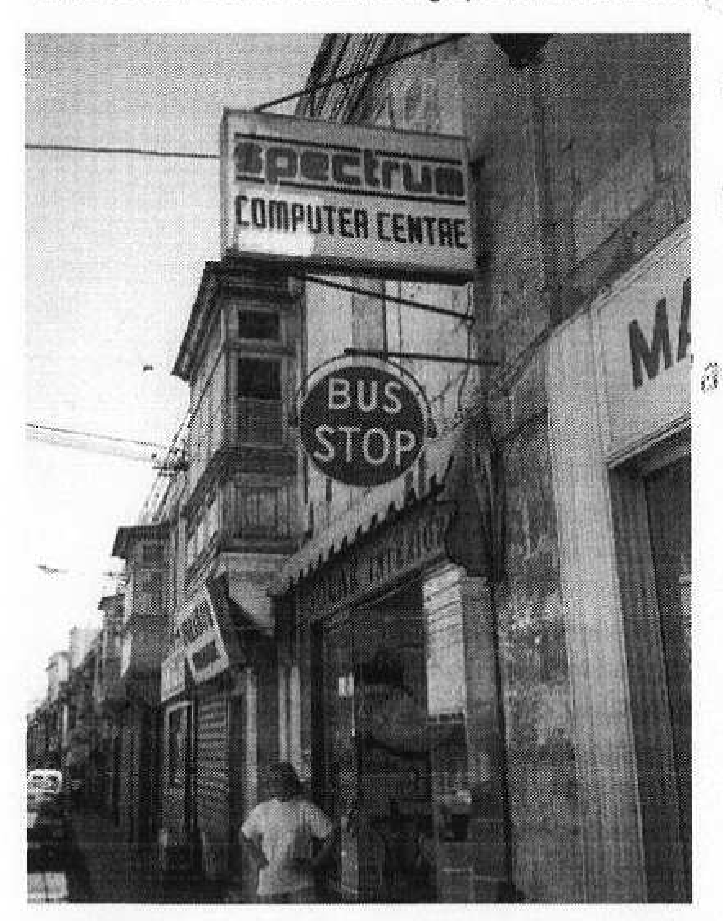


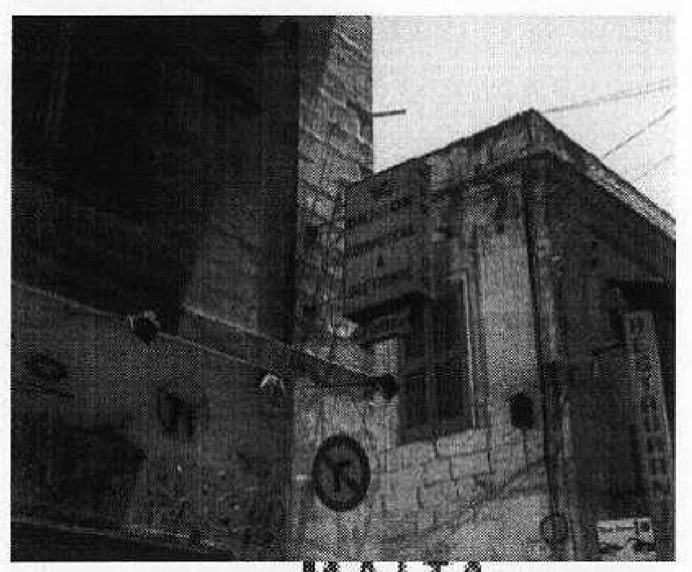


(Screens taken from Volga Soft „Love Gun")

Malta - auf den Spuren versunkener Computerkulturen

Eigentlich hätte ich mir den Urlaub auf Malta überhaupt nicht leisten können, aber mit einem kräftigen Zuschuss der ZX-TEAM Ortsgruppe Kiel und der Zusage von Wo, mir 2000.- DM Spesen zu ersetzen, ging es dann doch. Ich sollte mich allerdings auch mit der Frage befassen ob es auf Malta außer den normalen Flach- und Hochkulturen auch eine Computerkultur gegeben hat.
Ich bin bei meinen Forschungen auf sensationelle Dinge gestoßen. Das Inselreich Malta hat eine bewegte Geschichte, Römer, Araber, Türken, Engländer und Kölner waren schon auf den Inseln und haben ihre Spuren hinterlassen. Die jetzige Hauptstadt Valletta wurde vor 400 Jahren vom Ordensgroßmeister des Malteserordens Jean de la Vallette auf dem Reißbrett geplant und dann





gebaut. Und hier stoßen wir bereits auf die erste Spur einer versunkenen Computerkultur, der Name VALLETTA entstand aus den Wortteilen VAL und LET, zwei hinreichend bekannten Tastaturkürzeln des Spectrum, das AT am Ende erfuhr über die Jahrhunderte einen Dreher, das konnte ich im Archiv der Stadt nachlesen.
Auch die Nachbarinsel GOZO hieß vor etwa 300 Jahren noch GOTO, ein weiterer Beweis, daß der Spectrum seine Wurzeln auf Malta hatte. Tagelang habe ich nach weiteren Spuren und Beweisen gesucht, dann der größte Moment, andächtig kniete ich nieder. An der Außenwand eines historischen Gebäudes direkt an der alten Handelsstraße von Valletta nach Hamsun hing sie noch, die Tafel mit der Inschrift "Spectrum Computer Centre " .Das Centre gibt es nach Befragung ältester Einwohner schon über zwei Jahrhunderte nicht mehr, aber die alte Tafel mit der beweiskräftigen Inschrift ist geblieben (siehe Foto).
Und als ich dann zwei Tage später in einer Seitenstraße zum Großmeisterpalast noch eine weitere Tafel mit der Inschrift "DRAGON Computers & Software" fand, war der Erfolg der Reise gesichert. Die Fotos stehen Wolfgang zur Verfügung, die Abstammung des Spectrum ist endlich geklärt, das Gerücht, er wäre in England entwickelt worden, ist endgültig wiederlegt. Ich bedanke mich bei allen, die mir diese Reise ermöglichten.

Willi, Juni 2001

Eigentlich sauber recherchiert, Willi. Aber ich kann auch noch was dazu beitragen. Es gibt nämlich weitere Orte, die vom Spectrum zeugen und wo die Endung AT erhalten geblieben ist, z.B. Rabat (gleich zweimal, einmal davon auf Gozo, ehemals GOTO wie du richtig bemerkt hast), Sannat und Zabbat.
Einen wesentlichen Hinweis aber hast Du völlig ausser acht gelassen, weswegen ich von den 2000 DM Spesen auch wieder etwas zurückverlangen muss: Der Ort Hamrun weist doch eindeutig den Kürzel RUN auf, oder? Sowas darf einfach nicht passieren! Kontonummer ist Dir ja bekannt...
Leider hat die Zeit auch vor Malta nicht halt gemacht. Der PC zog ein, davon zeugt der Name Comino. Entstanden aus COM (1 oder 2 war nicht auszumachen), IN und dann das erstaunte O, wenn sich nix tat. Aus Zorn entschied man sich dann auch für eine andere Schreibweise: Kemmuna. Soweit meine Recherche. *Wo*

MONS - Befehle

(Kurzfasssung)

SS 3 „#“	zwischen hexidezimal/dezimal umschalten
SS 4 „$“	von Memory-Pointer disassemblieren
ENTER	Memory Pointer um eins erhöhen
CS 7 „^“	Memory Pointer um eins erniedrigen
CS 5 „<“	Memory Pointer um acht erniedrigen
CS 8 „>“	Memory Pointer um acht erhöhen
,	Adresse auf Stack wird Memory Pointer
G	String suchen. Der String wird durch Hexbytes+ENTER nach dem letzten Byte eingegeben.
H	Dezimalzahlen in Hexzahl umwandeln
I	Speicherblock kopieren (intelligenter Blockschutz)
J	Code von einzugebender Adresse ausführen
SS K	Code von Adresse PC ausführen
L	Speicherdump (zurück mit CS 5)
M	Memory Pointer setzen
N	Suche mit G fortsetzen
O	Memorydisplay nach dem Displacement von (Memory Pointer) setzen
P	Speicherblock mit BYtes füllen
Q	Registersätze austauschen
SS T	Breakpoint nach dem nächsten Befehl setzen und ausführen
T	Code disassemblieren
U	die *letzte* Veränderung von O rückgängig machen
V	die *letzte* Veränderung von X rückgängig machen
W	Breakpoint auf Memory Pointer setzen
X	Memory Pointer mit der Adresse von CALL oder JP updaten
Y	Ascii ab Memory Pointer eingeben
SS Z	Single Step. PC und Memory Counter *müssen* gesetzt sein.
SS P	Wie L, nur Ausgabe auf den Drukker

Speicher verändern

Eingabe eines Hexadezimalbyte bei „>“ modifiziert das Byte am Memory Pointer.
Direkt anschliessend können Memory Pointerbefehle folgen.

Register verändern

Hexadezimalbyte am Prompt „>“ gefolgt von „.“ verändert das mit „>“ markierte Register. Nächstes Register wird durch einfachen Druck auf „.“ ausgewählt.

Eingesandt von

Nele Abels-Ludwig
Friedrich-Ebert-Straße 36, 35039 Marburg
Abels@stud-mailer.uni-marburg.de



Faktor 0,983

So. 15.6.1986

Heute war mein Onkel Sepp zu Besuch. Er kam gleich nach dem Mittagessen, und ich wollte ihm schon sagen, daß er ja so um 15 Uhr wieder fahren könnte damit er rechtzeitig zum Kaffeetrinken wieder zuhause ist, aber mir fiel im letzten Moment ein daß er sich ja damals an meinem Spectrum beteiligt hatte, den meine Eltern mit zum 18. Geburtstag geschenkt hatten.

Der Spectrum ...... ist schon seltsam. ich hab ihn jetzt seit etwa 3 Jahren, und mache vieles mit ihm. Nur die Speicherei auf Kassette nervt, wird Zeit daß da mal was erfunden wird. Mein Freund Jello meinte mal, irgendwann wird einer auf die Idee kommen, dem Spectrum eine Festplatte zu verpassen. Ich grinste und sagte "komm von den Drogen runter" .... so ne Schnapsidee, das wird nie passieren! Aber der Spectrum und ich sind mittlerweile unzertrennlich, und ich bin immer auf der Suche was man ihm für Aufgaben geben kann.

Mein Onkel unterbrach meine Erinnerungen und fragte ob ich ihm nicht helfen könnte, kommende Woche seine Küche zu tapezieren. Aha, daher weht der Wind, Onkelchen .... naja er hat ja wirklich was gut bei mir. Da kam mir außerdem eine blendende Idee, sollte er doch mal sehen daß es ne gute Idee war, seinem Neffen den Computer zu schenken. Ich sagte :

"Alles klar, Onkel, Sepp, ich helf dir. Und ich mach das sogar allein, am kommenden Samstag. Ich werd keine Hilfe brauchen, weil ich mit dem Computer Berechnungen machen kann, und dann geht alles viel schneller." Mir schwebte vor, die Anzahl der benötigten Tapetenrollen anhand des Rauminhalts zu berechnen, die Länge der Bahnen auf den Millimeter genau auszudrucken usw.

Mein Onkel sah erst verwirrt, dann begeistert aus. Die Aussicht, nicht mithelfen zu müssen, war schon aufregend. So vereinbarten wir daß ich am kommenden Samstag 8 Uhr morgens bei ihm auftauche und er dann verschwinden kann. Werkzeuge, Tapetentisch, Leim etc wollte er bereitstellen. Außerdem würde ich 50 DM bekommen als Dankeschön - reicht für ein neues Spiel für meinen Specci, der soll ja auch was davon haben.....

Mi. 18.6.1986

Hmmm ob ich mich da übernommen habe? ich hab mich in den letzten zwei Tagen vor dem Specci mit diesen Berechnungen beschäftigt, und merke daß das doch etwas schwierig ist. Hmmm am besten ich mach dann die Berechnungen vor Ort, schließlich muß ich ja am Samstagmorgen die benötigten Tapeten noch kaufen, bevor die Geschäfte zumachen.

Do 19.6.1986

Die Zeit rast dahin, aber ich komm nicht gut voran .... bekomme langsam Panik. Eine Zehntelsekunde lang war ich versucht, Onkel Sepp anzurufen und das ganze abzublasen, aber .... neeeee ein Wort ist ein Wort!

Fr. 20.6.1986

Bingo !
in der Happy Computer 5/86 auf Seite 107 las ich heute einen Artikel über strukturiertes Programmieren, klar daß ich nicht vorankam mitmeinem Programm, das lag alles nur am Spaghetticode meines Basic-Programms.

Ich hab den Eindruck jetzt flutscht es wahnsinnig schnell. Innerhalb von 2 Stunden hatte ich mein Programm fertig, auf Tape gesichert. Eigentlich wollte ich ja jetzt noch eine Vergleichsrechnung machen, aber ich bin zu müde. Morgen *gähn*

Sa. 21.6.1986

Pünktlich um 8 Uhr stand ich bei meinem Onkel auf der Matte, und schleppte den Spectrum, einen kleinen Fernseher und meinen Kassettenrecorder in sein Wohnzimmer. Mein Vater hatte mal wieder gemault, aber gegen die moderne Computerrevolution kann man eh nix tun, sagt Jello immer. Recht hat er!

Gleich bei meinem Onkel in der Nähe ist ein Heimwerkermarkt, und so würde ich die Tapeten dann auch kaufen können, ohne ein Auto zu benötigen (nun bin ich schon 21 Jahre, und hab den Führerschein gemacht, aber Papa vertraut mir seine Karre immer noch nicht an ... naja .....)

Hab also an alles gedacht. Mein Onkel gab mir die 50 DM und Geld für die Tapeten, und verschwand mit meinem Papi, nicht ohne einen etwas mißtrauischen Blick auf meine Ausstattung zu werfen. Seis drum, der Respekt würde kommen wenn er diese Computerpräzise tapezierte Küche wiedersehen würde!

Rauhfaser ..... hmmmmm ich starte mein Programm, gebe Wandhöhe, Breite, Fenstergrößen als Abzug ein, und einen Faktor 0,983 Da bin ich superstolz drauf. Ich hab nämlich die Dehnung der Tapete berechnet! Wenn sie feucht wird, dehnt sie sich, also kann man beruhigt etwas abziehen von der Schnittlänge, tja, Computerpräzision.

Mann war ich stolz! Innerhalb etwa einer Viertelstunde hatte ich die Anzahl der Tapetenrollen und welche Bahn wie lang geschnitten sein muß. Das Basicprogramm (strukturiert, jawohl!) rechnete relativ fix. Ich bin also zum Markt, kaufte die Rollen, und fing sofort an die Bahnen zuzuschneiden. Ich vertraute natürlich meinen Berechnungen, die übrigens, falls da grade jemand grinst, wirklich absolut exakt waren.

Jede Bahn, feucht vom Kleister, reichte EXAKT vom Deckenansatz bis zur Zierleiste, wahnsinn. Im Traum sah ich mich schon in der Happy Computer den Programmierwettbewerb gewinnen, und tapezierte richtig fleissig weiter.

Am späten Nachnmittag war ich fertig. Fix und fertig. Ein letzter Blick, jawohl, die Bahnen waren eine Augenweide. Millimetergenau bis zur..... moment, hier an der zuerst geklebten Bahn war unten ein Millimeter zuwenig ..... hmmmmm naja das stört ja wirklich nicht. Ein Blick nach oben an die Decke zeigte daß auch da etwas fehlte. Seis drum ,war in der Ecke. Ich kippte ein Fenster zum lüften, setzte mich ins Wohnzimmer von meinem Onkel und gönnte mir eine schöne Pause. Ich probierte gerade ein neues Spiel aus als mein Onkel zurückkehrte. Stolz führte ich ihn in die Küche, und wollte ihm mein Werk zeigen. Auf dem kurzen Stück vom Flur bis dahin erzählte ich dauernd von der Präzision, mit der der Spectrum das alles ermöglicht hat.

Dann kamen wir in die Küche, und mir schürte es den Magen zu! ALLE Bahnen waren etwa einen Zentimeter unten und oben zu kurz!!! Mein Onkel sagte mir sofort daß ich wohl nicht daran gedacht habe, daß Tapeten sich nicht nur dehnen, wenn sie feucht werden, sondern auch zusammenziehen wenn sie trocknen!! Au sch.....

So. 22.6.1986

Mein Onkel war wieder zu Besuch bei uns. Ich fühlte mich elend, denn wir würden kommende Woche alles nacharbeiten müssen. Mein Onkel nahm es aber mit Humor, nur ein Satz von ihm war etwas sarkastisch:

Jeder Tapetenbahn fehlte dank der Computerpräzision genau dieselbe Länge.....
Faktor 0,983 !

Einiges zum Lunter Emulator

In verschiedenen Beiträgen, letztlich im SPC Heft Mai 2000, Seite 12 berichtete ich zum Lunter Emulator. Bei der Nutzung des Emulators unter Windows 95 mußte ich feststellen, daß einige wenige der vielfältigen darin enthaltenen Optionen nicht wunschgemäß funktionieren.

Das betrifft vorwiegend einige Funktionen im +D Modus : Wenn man versucht, mit "Format D1" eine Diskette zu formatieren, erhält man lediglich eine Fehlermeldung "CHECK DISK in Drive" oder "FORMAT Data lost".

Auf eine diesbezügliche Frage in einem Brief an Ron Cavers, dem jetzigen und einzigen Distributor des Lunter Emulators, erhielt ich die Antwort, daß es für eine Diskettenformatierung in "pure DOS" keine Probleme geben sollte. Ich hatte bisher jedoch keine Erfolge, auch wenn ich den Computer im MS DOS gestartet hatte. Vielleicht gibt es doch Möglichkeiten, solch ein "pure DOS" zu starten, wo eine Formatierung funktionieren soll?

Da die Funktion für eine Bit zu Bit-Kopie mit dem Befehl "FORMAT D1 to 2" auch einen Formatierungsvorgang beinhaltet, ist deshalb auch das Direktkopieren von Disketten nicht möglich. Wer kein reales Spectrumsystem mit +D Interface hat, der sollte sich also vorsorglich formatierte Disketten beschaffen.

Ron Cavers gab in diesem Zusammenhang noch folgenden Hinweis: " but you can create copies of your Spectrum disks by using the DOS program called "TELEDISK". This creates an image file of almost any floppy disk, whatever the format and stores it on your harddrive. You can than run the program again and use the image file to create an exact copy of your original disk. Any search of the Internet should find a copy of this popular program."

Das wäre also auch ein Weg, +D Disketten mit dem PC zu kopieren und so auch Leerdisketten herzustellen, indem man eine vorhandene Leerdiskette kopiert. Wer hängt am Internet, kann das Progranun " TELEDISK" aufstöbern und ergründet diese Kopiermöglichkeit?

Weiterhin: Versuche, nach einem NMI und Drücken von 1 oder 2, mit einem 8- oder 24-Nadeldrucker, F 80 oder LQ 400, am Parallelport des Computers angeschlossen, einen Ausdruck im 32 Spalten- oder High Res-Format zu machen, führten zu keinem Erfolg.

Ebenfalls war es mir nicht möglich, im Multiface 128-Modus, der ja auch im Emulator enthalten ist, mit dem Befehl ,'PRINT' einen Ausdruck mit oben angeführten Druckern zu erhalten.

(Hinzufügend sei hier noch bemerkt, daß meines Wissens nach der Lunter Emulator der einzige der bekannt gewordenen Emulatoren ist, der das +D Disketteninterface, das Multiface MF 128 und noch weitere Hardwarezusätze nachbilden kann)

Diese Druckersituation ist zwar unbefriedigend aber sonst kein schmerzlicher Nachteil weiter. Es wird heutzutage doch kaum jemanden geben, der im Emulatormodus Texte in z.B. Tasword oder TLW verfasst und solche oder auch andere ausdrucken will. Eine Textbearbeitung mit Spectrumprogrammen ist wohl den Computerfreunden vorbehalten, die mit dem realen Spectrumcomputer arbeiten. PC Besitzer verwenden da eines der üblichen Textprogramme, wie z.B. MS Word oder Wordpad.

Um Screens von Spectrumprogrammen mit dem PC auszudrucken, ist z.B. der Ausdruck aus einem in ein übliches Bildbearbeitungsprogramm eingegebenen .GIF Bildes möglich, das mit Hilfe des Emulators erstellt oder aus einem Screen-Snapshot, der in ein .GIF-Format umgewandelt wurde. Man kann auch

ein mittels der Taste "Druck" in den Zwischenspeicher geladenes und in ein Bildbearbeitungsprogranun gebrachtes Bild drucken. Hier hat man noch den Vorteil, daß man jeden modemen Farbdrucker verwenden kann, der mit dem PC arbeitet.

Und: Wer hat noch einen der früheren, heute nicht mehr für die nichtprofessionelle Nutzung handelsüblichen Nadeldrucker effektiv in Betrieb? Meist Spectrumfreunde, die noch ihren 8-Bitter als Hobby betreiben.

Allgemein ist noch zu sagen, daß es einige wenige Programme gibt, die mit dem Emulator nicht erfaßt werden können. Das wohl wegen Konflikten mit Interrupt- oder Di-Befehlen. Das betrifft z.B. die Programme "LAST WILL DIZZY" und das MGT Boot Programm. Speziell bei der Erprobung unter Windows 98 gibt es einige zusätzliche Probleme. Bei Versuchen unter Windows 98 verwendete ich den gleichen PC und tauschte nur die Systemfestplatte mit W95 gegen eine mit dem Betriebssystem W98 aus. Hier funktionierten im +D Modus nach dem NMI auch die Befehle "SNAP SCREEN" nach Drücken von 3, "SNAP PROGRAM 48 K" nach Drücken von 4 und "SNAP PROGRAM 128 K" nach Drükken von 5 ebenfalls nicht. Außerdem erhält man hier unter W 98 nach dem Löschbefehl "ERASE D1 Filename" und auch nach dem Speicherbefehl "SAVE D1 Filename" nur eine Meldung "DISK WRITE PROTECTED" (Schreibschutz natürlich aufgehoben). Unter Windows 98 gibt es auch noch einige weitere Programme, die sich nicht einladen lassen, wie z.B. "THE COPIER" oder "PAPERBOY". Auf die Anfrage an Ron Cavers zu den WIN98 -Problemen konnte er hierzu keine Auskunft geben, da er hierzu keine Erfahrung vorläge.

Das Nichtfunktionieren der hier aufgeführten speziellen Befehle, die nicht erwartungsgemäß arbeiten, ist zwar nicht allzu gewichtig. Es zeigt sich jedoch, daß durch die Windows-Weiterentwicklung auch eine solche des Emulators erforderlich ist, wenn er unter diesem System problemlos arbeiten, existenzfähig und aktuell sein soll.

Die 1999 stattgefundene Fortentwicklung des Lunter Emulators durch die Ergänzung WINZ80 bringt zwar einige weitere Features, beseitigt aber die neuen Probleme und die zum +D nicht. Zu WINZ80 ist noch zu bemerken, daß es einige auf CD's angebotene früher gemachte Snaps gibt, die zwar unter dem Emulator Z80 laufen, aber unter WINZ80 nicht. Da kommt mitunter eine Meldung, daß ein ungeeignetes Format vorhanden ist.

Hier gibt es die Abhilfe, unter Z80 einen Snap von dem Programm zu machen. Dieser läßt sich dann auch unter WINZ80 einladen. Betroffen von diesem Makel sind meist früher gemachte Programmsnaps mit der Dateierweitenmg .SNA.

Auf die weitere Anfrage an Ron Cavers zu den weiteren Aussichten für den Emulator, insbesondere, ob Weiterentwicklungen beabsichtigt sind, schrieb er: "Regarding the feature of the Emulator I don't know what Gerton Lunter's planes are. (He is away at the moment, having just completet his thesis)" Das heißt, er ist zur Zeit mit andren Dingen beschäftigt, wohl mit seiner Doktorarbeit. Wir müssen also abwarten, ob es weitere an die neueren Windows-Systeme wie auch Windows 2000 oder Windows ME angepasste Versionen zum Emulator geben wird. Ich persönlich behalte weiterhin das Windows 95, wo es z.Zt. die wenigsten Behinderungen gibt.

Auf meine Frage, ob der Emulator als Shareware mit Registrienmgsmöglichkeit ggf. Freeware wird, bekam ich die Antwort: "There are still no immediate planes for it to be freeware". Warum ich meine Fragen an Ron Cavers stellte hat den Grund, daß ich einen Brief an Gerton Lunter an seine bislang bekannte Adresse als unzustellbar von der Post zurück erhielt.

Das sind nun die letzten Meldungen zum Lunter Emulator, die ich im März 2001 von Ron Cavers erhielt.

Vielleicht gibt es Leser, die einen Internetanschluß haben und ggf. Verbindung mit G. Lunter aufnehmen können.

Diesmal habe ich mehr zu einigen aufgekommenen Fragen zum Emulator berichtet Wir können nur hoffen, daß dieser Klassiker doch noch einige "Auffrischungen" erhält, um auch weiterhin zufriedene Nutzer zu haben.

Zum Abschluß noch einige Tips zum Lunter Emulator:

In den Eigenschaften zur Datei Z80.exe sollte man in der Registerkarte "Speicher" in alle Eingabefelder für Speicherbereiche "automatisch" eingeben, wenn es beim Starten des Programms zu Meldungen zu Speicherbereichen geben sollte.

In der Registerkarte "Programm" kann man beim Startendes Emulators Z80 den 8 Sekunden lang gezeigten Screen für die Lunter Emulator- Versionsinformation unterdrücken, wenn man in der Befehlsteile nach ...\ Z80. EXE noch als Schalter anhängt: Ein Leerfeld, Bindestrich und zweimal Null.

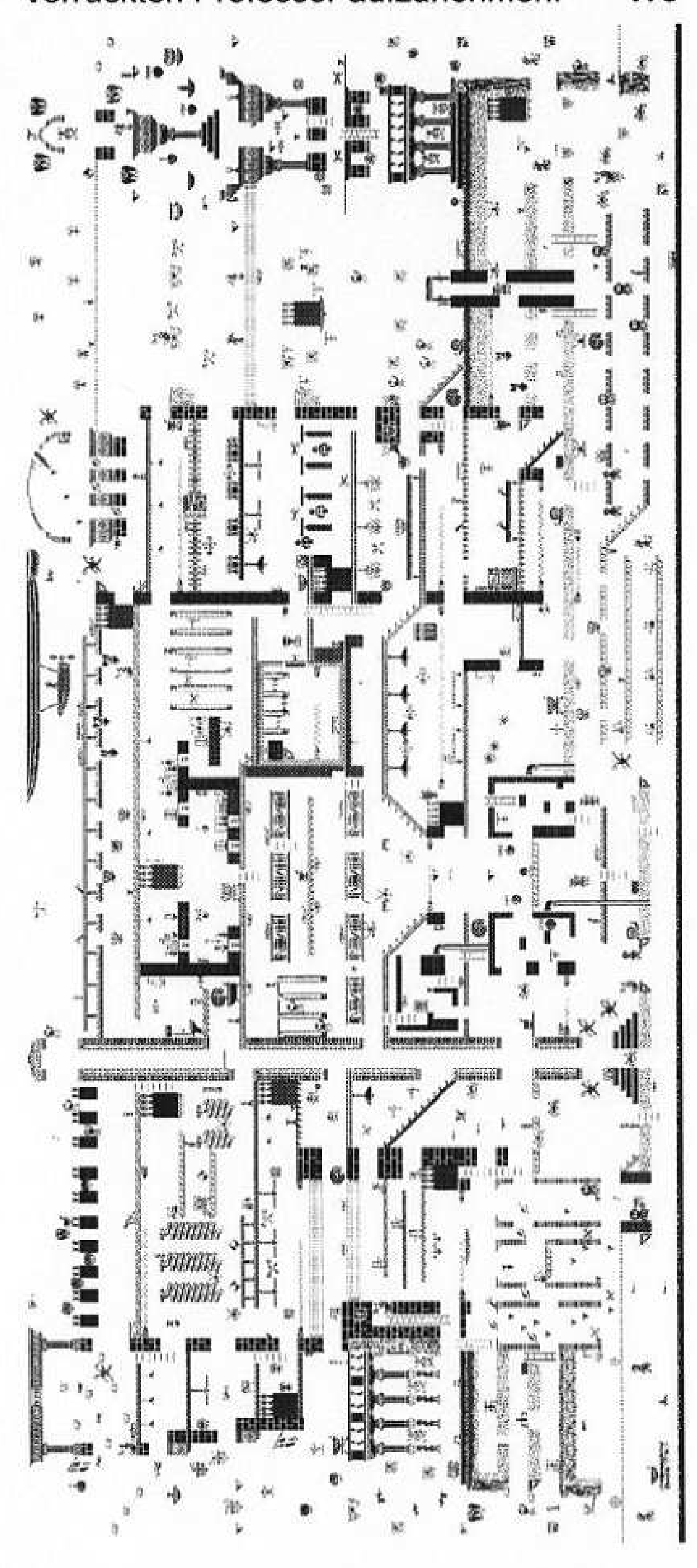
Damit beim Beenden des Emulators Z80 (F10,Q,Y) nicht erst noch eimnal das extra zu beendende DOS-Fenster erscheint, ist in der Registerkarte "Programm" ein Häkchen in "Beim Beenden Schließen" einzubringen.

Und ganz zum Abschluß möchte ich nochmals alle Clubfreunde und Leser bitten, wenn Sie etwas zu oben geschilderten Dingen sagen können, das, und wenn es auch die kleinste Mitteilung ist, an WOMO zu übermitteln. Manchmal sind auch die unwichtig scheinenden Dinge für andere wertvoll und wissenswert.

Heinz Schober
Taubenheimer Straße 18
01324 Dresden, im Juni 2001

Dynamite Dan

Einer der grossen Hits unter den Spielen der damaligen Zeit war Mirrorsofts „Dynamite Dan". Vor längerer Zeit habe ich diesen Plan dazu (irgendwo) im Internet gefunden. Vielleicht lädt er ja ein, den Kampf gegen den verrückten Professor aufzunehmen. *Wo*





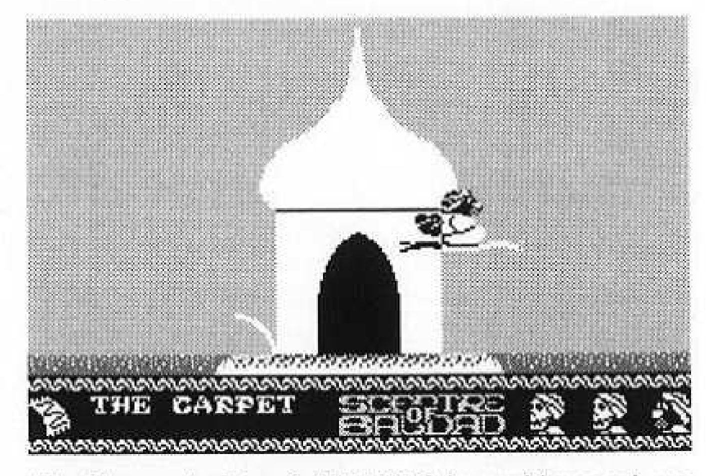
Hallo Spielefreunde!!

Dieses Mal hat uns die Suche nach einer neuen Herausforderung weg von den unsererseits so beliebten Textadventures und hin zu einem Grafikadventure geführt. Daß gute Programme nicht unbedingt immer teuer sein müssen (jedenfalls in Relation zu ihrem Ersterscheinen), das konnte man schon 1987 sehen, als das Programm, dem wir uns heute näher widmen wollen, von Atlantis Software Ltd. auf den Markt gebracht wurde. Wie sich vielleicht schon einige von euch denken können handelt es sich um den ASM-Hit **„The Sceptre of Bagdad"**, einem Low-Budget Spiel das allerdings viele in seinen Bann gezogen hat. So auch uns. Wer nochmal die Kritik/Bewertung nachlesen will, sie findet sich in der Ausgabe 9/87 der Zeitschrift ASM auf Seite 43.

Doch worum geht es eigentlich? Nun die Hintergrundgeschichte ist schnell erzählt und die Aufgabe hört sich auch nicht unbedingt sehr schwierig an, was aber vielleicht anfangs etwas täuscht. Wie der Name schon sagt, spielt das Programm in der Hauptstadt des Irak in Bagdad (lange Zeit vor den Gewaltherrschern heutiger Tage). Der Spieler verkörpert hierbei den Kalifen von Bagdad, dessen Aufgabe es ist, sich am Neujahrstag seinem Volk mit dem Zepter auf dem Balkon seines Palastes zu zeigen. Das ist allerdings leichter gesagt als getan, denn unser Zepter ist momentan nicht auffindbar und auch in unserem Reich scheint einiges merkwürdig. Der Kalif (und damit der Spieler) hat bis Mittag Zeit sein Zepter zu finden und auf dem Balkon zu erscheinen. Somit wissen wir, was zu tun ist. Das Programm hat sich unserer unbedeutenden Meinung nach nicht umsonst den ASM-Hit geholt, denn es ist erstens grafisch sehr schön gestaltet und auch ausserordentlich gut spielbar, was in unseren Augen noch wichtiger ist. Wenn man sich nicht zu ungeschickt anstellt und ein bischen überlegt handelt, ist es kein Problem mit den drei zur Verfügung stehenden Leben durchzukommen.

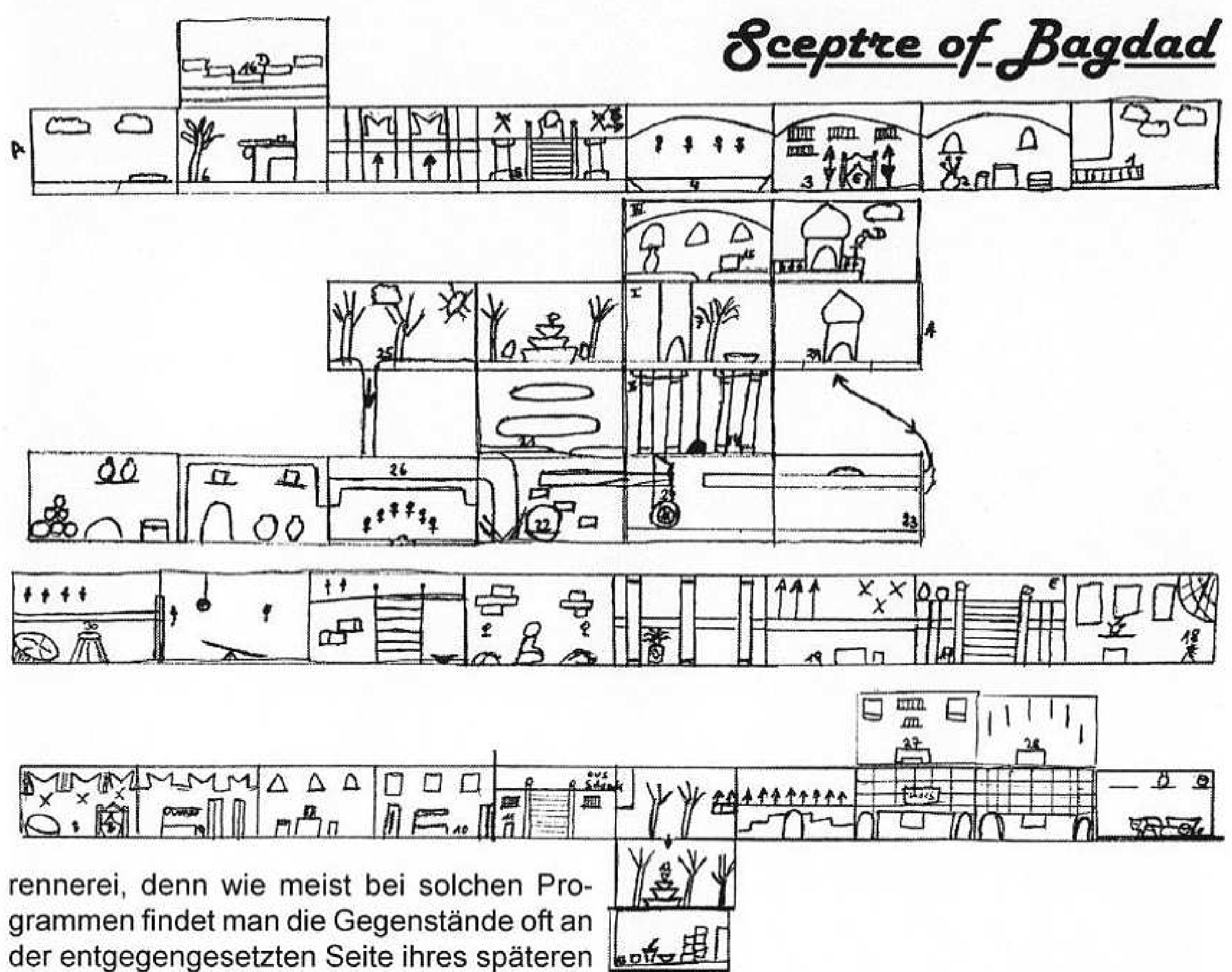
Wie es sich für Spiele dieses Genres versteht, gibt es eine Menge an Gegenständen, mit denen man herumhantieren kann um sie so einzusetzen, daß einem ein Weiterkommen sicher ist. Daß man dabei nicht immer auf die Logik setzen darf, sei hier nur am Rande erwähnt. Gelegentlich kann man keinen direkten Zusammenhang zwischen Gegenstand und Wirkung herstellen, aber das macht die Knobelei manchmal noch ein bischen spannender.

Der Komplex in dem sich das Programm abspielt (also der Palast und die angrenzenden Örtlichkeiten unseres Reiches) ist nicht unüberschaubar groß. Einen Überblick könnt ihr euch im beiliegenden Plan verschaffen. Allerdings erfordert es einer ziemlichen Rum-



Ein fliegender Teppich!!! Weiß irgend jemand, wo man dieses prächtige Verkehrsmittel bekommen kann...?

Sceptre of Bagdad



rennerei, denn wie meist bei solchen Programmen findet man die Gegenstände oft an der entgegengesetzten Seite ihres späteren Einsatzes. Sicherlich ließe sich zu diesem Programm ein einheitlicher Lösungsweg darstellen. Doch ist dies in unseren Augen nicht notwendig, denn man kann das Spiel auf mehrere Arten lösen. Manche Handlungsabfolgen sind jedoch unabänderlich und drängen sich eigentlich sofort logisch auf. Aber ihr sollt ja auch noch was zum Knobeln haben. Ein weiterer Grund dafür an dieser Stelle keinen fertigen Lösungsweg zu veröffentlichen war der, daß der eine oder andere von euch ja vielleicht schon ein schönes Stück des Weges alleine geschafft hat, und nur an einer Kleinigkeit hängt. Deshalb die nachfolgenden Erläuterungen zu den Gegenständen und ihren Wirkungen. Damit kann sich jeder heraussuchen was er braucht ohne gleich den gesamten Lösungsweg zu kennen. Allerdings kann man unter Beachtung der Tips das Programm vollständig lösen. Noch ein Hinweis bevor wir die Sache angehen. Manche Räume kann man nur betreten, wenn man einen bestimmten Gegenstand dabei hat. Ist dies einmal geschehen, kommt man auch ohne diesen Gegenstand wieder dort hin. Aber keine Regel ohne Ausnahmen. Eine davon ist die Geheimtür, die sich imner nur mit dem Ali Baba Book öffnen läßt. Die Wüste ist auch ein solcher Ort.

Denkt bitte an die Kokosnuss. Um schließlich an den Spiegel zu kommen, benötigt man den Fish Helmet (er schützt uns vor dem Ertrinken). Der Helm darf aber nicht gegen den Spiegel getauscht werden. Also als zweiten, sozusagen Tauschgegenstand, etwas mitnehmen, das nicht mehr benötigt wird oder auch keinen Nutzen hat (siehe nachfolgende Aufstellung) und darauf achten, daß dieser Gegenstand aktiviert (angewählt) ist.

Doch kommen wir nun zu unseren Erläuterungen und Spielhilfen. In der nachfolgenden Liste sind zwei Gegenstände nicht aufgeführt, da sie der Spieler bereits beim Programmstart bei sich hat. Es sind dies der „Miner Hat" und das „Hanky" .

Ersterer dient dazu in dem Zimmer mit dem Spiegel an der resoluten Lady vorbei zu kommen. Mit dem Taschentuch sind wir in der Lage die im Startraum stehenden Lampe zu reiben, um ihr verschlüsselte Informationen zum Spiel zu entlocken. Alle anderen Gegenstände sind mit ihren Nummern im Plan eingezeichnet. Nachfolgend könnt ihr nunmehr lesen, um was es sich dabei handelt und wozu es dient.

The Whip (1) hilft einem um an dem Ochsengespann vorbei zu kommen, das sich ganz rechts im Plan befindet. Sonst kommt man nicht an die Achse heran, die benötigt wird um die Zugbrücke zu reparieren.

The Flute (2) dient uns um uns als Schlagenbeschwörer zu betätigen. Nachdem wir mit dem Teppich nach links geflogen sind, kommen wir zu einem Turm in dessen Inneren ein Seil liegt. Wenn wir die Flöte dabei haben steigt es in die Höhe und wir können daran hinaufklettern und weitere Erkundigungen anstellen bzw. weitere wichtige Gegenstände erreichen.

The Sword (3) dient dazu, im Raum mit der Spinne den Faden des Spinnennetzes abzuschneiden und ihn dann als Bogensaite zu benutzen.

The Trident (4) wird benötigt um der Muschel ganz unten beim Wasserfall eine Perle zu entlocken.

The Whisk (5) -Rechts ausserhalb des Palastes gibt es einen Brunnen mit einem Weg nach unten. Dort befindet sich ein Zauberer sowie zwei Gefäße in denen Feuer brennt. Wenn wir unbeschadet an die Lampe kommen wollen, müssen wir den Whisk dabei haben.

The Keys (6) helfen uns ganz rechts im Schlafzimmer durch den Schrank zu kommen.

The Coconut (7) benötigen wir als Lebenselexir um die Wüste unbeschadet zu durchqueren.

The Insect Net (8) -Damit fängt man die Biene ein, die erscheint, wenn man links ausserhalb des Palastes das Bienennest unterhalb des Daches losgetreten hat. Auf das Dach kommt man nur über den linken Turm und mit Hilfe des Vogels. Wenn das Insektennetz voll ist, vertreibt es die Spinne und wir können an den Faden gelangen.

The Fire Ring (9) -Ganz unterhalb des linken Brunnens- links von der Muschel ist eine Falltüre, die sich nur mit dem Ring öffnen läßt.

The Purse (10) -Vorerst noch leer. Jedoch kann man sie in einem Geheimraum (ziemlich links unten) mit Geld füllen und dann in den Läden rechts aussen einkaufen.

The Ali Baba Book (11) -Damit öffnet man die Geheimtür links von der Falltüre und kommt in den Raum mit den Geldstücken.

The Sling (12) wird benötigt, um die Kokosnuss vom Baum zu holen. Das ganze funktioniert aber erst, wenn sie mit der Perle als Projektil geladen wurde.

The Lamp (13) braucht man erst kurz vor Spielende um am Sprungbrett hinter dem Zyklopen weiter zu kommen. (Der Bogen muß auch dabei sein).

The Hour Glass (14) wird nicht benötigt.

The Feather (15) -Damit überredet man den Vogel oben auf dem Turm zu einem Rundflug.

Several Wings (16) veranlassen die Statue, die auf dem linken Brunnen steht, sich in die Lüfte zu erheben und uns den Weg nach unten freizugeben.

The Carpet (17) wird nicht benötigt.



The String (18) dient als Bogensaite. The Armour (19) wird nicht benötigt.

The Whole Axle (20) -Damit repariert man die defekte Zugbrücke unterhalb des linken Brunnens.

The Quiver (21) braucht man um den Bogen zu laden.

The Pearl (22) -Dient als Munition für die Schleuder.

The Fish Helmet (23) dient als Tauchhelm um uns unbeschadet in dem Wasserlauf unter der Wüste treiben lassen zu können.

The Broken Axle (24) wird nicht benötigt.

The Trunk Blunt (25) wird später, nachdem er mit der Axt geschärft wurde dazu benutzt, den Zyklopen zu vertreiben.

The Mirror (26) -Damit bannen wir die Medusa im Raum links von der Rüstung.

The Slippers (27) -Damit können wir die glühenden Kohlen unterhalb des linken Brunnens (gleich nach der Zugbrücke) gefahrlos überqueren und gelangen so in den Turm, den wir sonst immer nur im Flug gesehen haben.
The Axe (28) -Damit schärfen wir den Trunk Blunt bevor es zum Zyklopen geht.

The Bow Unstrung (29) -Ein Teil um einen Bogen zu basteln.

The Sceptre (30) -Endlich gefunden. Damit zeigen wir uns auf dem Balkon des Palastes und haben das Spiel gelöst.

Ja Leute, so schnell kann es gehen. Das war alles, was es über unsere Reisen in den Nahen Osten zu berichten gab. Wir denken es war ein schönes Spiel, von deren Art es mehr geben könnte. Aber da sind ja noch einige Programme die von uns gelöst werden wollen. Wartet ab was es in einem der nächsten Infos zu lesen gibt. Bis dann.......

Harald R. Lack
Heidenauer Str. 5, 83064 Raubing
Hubert Kracher
Schulweg 6, 83064 Großholzhausen

Expertenstreit

Die Experten streiten sich darüber, ob Computer männlichen oder weiblichen Geschlechts seien.

Die Frauen glauben männlich:

- Er hat jede Menge Wissen, ist aber trotzdem planlos.
- Er soll helfen Probleme zu lösen, ist aber die halbe Zeit selbst das Problem.
- Sobald man sich einen zulegt, kommt man darauf, dass wenn man ein bißchen gewartet hätte, ein besserer zu haben gewesen wäre.

Die Männer hingegen halten den Computer für weiblich:

- Nicht einmal der Schöpfer versteht ihre innere Logik.
- Die Sprache, mit der sie sich untereinander verständigen, ist für niemanden sonst verständlich.
- Sogar die kleinsten Fehler werden im Langzeitgedächtnis zur späteren Verwendung abgespeichert.
- Sobald man einen hat, geht das ganze Geld für Zubehör drauf.

Gefunden in der Joyce-Klubzeitung Nr. 61

"Spreadsheet als one-liner"

Wie bekannt gefallen mir ONELINERS sehr. Oneliners sind Programme, welche nur eine Zeile Code besitzen. Jetzt habe ich sogar ein Spreadsheet als ONE-LINER kodiert. Dieses Spreadsheet kennt Formel und Texteingabe, Load- und Savemöglichkeit, Cursorkeys zur Steuerung, ENTER für Formeleingabe und SPACE zum Beenden. Den Wert eines 'Feldes' kann man mit FN W(1,1) fur 1,1 abfragen usw....

Formelbeispiel in 1,1:
    FN w(1,2)*4: Resultat = -400
Formelbeispiel in 1,2:
    -100: Resultat = -100
Formulbeispiel in 1,3 :
    !Tekst : Resultat = Tekst

Der Code ist hierunter. Dann folgt den Beschreibung.

```
   1 DEF FN w(a,s)=VAL ((a$(a,s)
 AND a$(a,s,1)>"!")+("0" AND a$
(a,s,1)<="!")): LET x=1: LET y=1:
 DIM b$(32): DIM a$(21,4,32): IN
PUT "LOAD?";n$: FOR h=1 TO n$>""
: LOAD n$ DATA a$(): NEXT h: CLS
 : FOR h=0 TO 1: FOR f=1 TO 21:
PRINT AT f,0;f;: FOR g=1 TO 4: P
RINT AT 0,g*7;g;AT f,g*7-3;((STR
$ FN w(f,g)+b$ AND a$(f,g,1)>"!
")+a$(f,g,2 TO )+b$)( TO 32-g*7+
3) AND a$(f,g,1)>" ": NEXT g: N
EXT f: FOR s=0 TO 1: PRINT OVER
1;AT y,x*7-3;"#######";#1;AT 0,0
;a$(y,x): PAUSE s: NEXT s: LET k
=CODE INKEY$: FOR j=1 TO k=13: I
NPUT a$(y,x): NEXT j: LET x=x+(k
=56)*(x<4)-(k=53)*(x>1): LET y=y
+(k=54)*(y<21)-(k=55)*(y>1): LET
 s=k*(k<48)-1: NEXT s: LET h=k=3
2: NEXT h: INPUT "SAVE?";n$: IF
n$>"" THEN SAVE n$ DATA a$()
```



Beispiel von Johan Koelman im Programm

Für die Experten:

```
DEF FN w(a,s)=VAL ((a$(a,s)
AND a$(a,s,1)>"!")+("0" AND a$
(a,s,1)<="!"))
```

Berechnung des Feldes. Wenn das Feld leer ist, wird der Wert 0 sein.

```
LET x=1: LET y=1: DIM b$(32): DIM
a$(21,4,32)
```

Variablen, A$ ist 'worksheet'

```
DIM b$(32): DIM a$(21,4,32): IN-
PUT "LOAD?";n$: FOR h=1 TO n$>""
: LOAD n$ DATA a$(): NEXT h: CLS
```

Loadroutine für bestehendes Worksheet. Wenn N$ eingegeben ist, dann wird ein vorhandenes Worksheet geladen.

```
FOR h=0 TO 1: FOR f=1 TO 21:
PRINT AT f,0;f;: FOR g=1 TO 4: P
RINT AT 0,g*7;g;AT f,g*7-3;((STR
$ FN w(f,g)+b$ AND a$(f,g,1)>"!
")+a$(f,g,2 TO )+b$)( TO 32-g*7+
3) AND a$(f,g,1)>" ": NEXT g: NEXT
f
```

Bildschirmroutinen, H-Zähler ist Repeat-loop für nächste Berechnung.

```
FOR s=0 TO 1: PRINT OVER
1;AT y,x*7-3;"#######";#1;AT 0,0
;a$(y,x): PAUSE s: NEXT s
```

Warteschleife, Cursor wird abgedruckt, auch die Formel im unteren Teil des Bildschirms,

```
 1) SPREADSHEET    23) SPOORZOEKER
 2) GETALRADEN     24) SCHIETTENT
 3) RUIMTEREIS     25) COLORIS
 4) MASTERMIND     26) BOUNCER
 5) GALGJE         27) CATCH
 6) MEMORY
 7) METEORSTORM
 8) BLOKJESDODER
 9) TIJDRACER
10) REBOUND
11) MIJNENVELD
12) MUZIEKSPELER
13) SLANG
14) FROGGER
15) BINGO
16) PACMAN
17) SCHUIFPUZZEL
18) TORENS VAN HANOI
19) US BOWLING
20) GROTREIS
21) TANKSLAG (2 PERS)
22) DOOLHOFGENERATOR

MAAK KEUZE ■
```

Johans Oneliner Programmsammlung

danach PAUSE 0, wenn ein Taste gedrückt ist, wird der Cursor gelöscht.
N.B. ##### sind in Grafik-MODE Shift 8

```
LET k=CODE INKEY$: FOR j=1 TO k=13: INPUT a$(y,x): NEXT j: LET x=x+(k=56)*(x<4)-(k=53)*(x>1): LET y=y+(k=54)*(y<21)-(k=55)*(y>1)
```

Controls: Mit der ENTER-Taste wird eine Eingabe abgefragt.
Einen Text anfangen mit einem "!"
Cursor-Tasten für Steuerung.
Formel kopieren mit EDIT und dann a$(1,2) für Kopie von W(1,2).
Wert kopieren mit EDIT und dann STR$ FN W(1,2) für Wert W(1,2)

```
LET s=k*(k<48)-1: NEXT s
```

Unübliches Nutzen einer FOR-NEXT-Schleife. Mit S-Zähler wird Cursor gedrückt, aber wenn nicht ENTER oder SPACE gedrückt ist, dann geht das Programm zurück in die FOR-NEXT Schleife. !!!!!!

```
LET h=k=32: NEXT h: INPUT "SAVE?";n$: IF n$>"" THEN SAVE n$ DATA a$()
```

Wenn Taste ist nicht SPACE, nächste Berechnung. Ist es aber doch SPACE, Frage zum Speichern des Worksheets. Nur wenn LEN N$ großer als 1 ist wird gespeichert.

Johan "Dr BEEP" Koelman

Colour Screendump

für Epson-kompatibele Tintenstrahl-Farbdrucker.

In letzter Zeit bin ich damit beschäftigt, eine PC-Tastatur und Mausschnittstelle für den Spectrum zu machen. Weil das gar nicht so einfach ist wie man denkt, und viel Zeit nimmt, brauchte ich etwas zur Zerstreuung. Dadurch kam ich zum Schreiben einer Colour Screendump Routine, weil diese relativ einfach ist. Und obwohl viele Leute sich mit Screendump Routinen beschäftigt haben, habe ich bis jetzt einen Farbdump noch niemals gesehen.
Weil ich selber ein Canon BJC4300 besitze, habe ich mir für diesen Drucker ausgesucht, wie man damit in Farben drucken kann. Der Canon ist Epson kompatibel, das bedeutet das man zuerst die Farbe wählen muss, und dann ein normales 'bit-image' Kommando gibt, wie z.B. das bekannte 'ESC "K" Anzahl'. Der Kommando für Farbwahl ist 'ESC "r" n' (CHR$ 27;"r";CHR$ n;) wo n die Werte von 0-6 haben kann (Schwarz, Magenta, Cyan, Blau, Gelb, Rot, Grün).
Der Spectrum arbeitet mit den Grundfarben Rot, Grün and Blau (RGB), wie wir alle wissen, aber der Canon Drucker arbeitet mit die Grundfarben Cyan, Magenta, Gelb und Schwarz (CMYK). Wie übersetzt man dass?

Schauen wir uns die Farben mal an.

	Speccy INK/ PAPER	Speccy Binär	Canon ESC "r" n	Canon Binär	Binär lsb zuerst
Schwarz	0	000	0	000	000
Blau	1	001	3	011	110
Rot	2	010	5	101	101
Magenta	3	011	1	001	100
Grün	4	100	6	110	011
Cyan	5	101	2	010	010
Gelb	6	110	4	100	001
Weiss	7	111	-	-	-

Da seht man keine Ähnlichkeit, oder doch? Setzen wir die Canon Bits in eine andere Reihenfolge. Zuerst das lsb (least significant bit), dann bekommt man die letzte Spalte. Ausgenommen für Schwarz sind die Bits invertiert (0->1, 1->0) im Vergleich zu den Bits in der Speccy Binär Spalte. Das invertieren ist sehr einfach in Maschinensprache zu übersetzen, dafür gibt es den XOR (eXclusive OR) Befehl.

Wie bekommt man die Farbe eines Pixels? Das ist einfach, man berechnet die Adresse des Pixelattributes und wenn das Pixel gesetzt ist, benutzt man die Inkfarbe, sonst die Paperfarbe.

Weil der Canon mit vier Grundfarben arbeitet, habe ich mich entschlossen für jede Farbe einen Puffer zu benutzen. Nach der Übersetzung schiebt man jedes Farbebit in seinen eigenen Puffer. Nach jeder Zeile von 32 Charaktern sendet man die Puffer an den Drucker bis man alle 24 Zeilen gehabt hat.

Die folgende Routine ist nicht sehr kurz, es braucht 1kB Puffer, aber ich lass das Verkürzen gerne Johan Koelman.

PS: Ich suche für das Keyboardinterface eine Firma, worüber ich das 74HCT22106 IC bekommen kann (wie im SAM keyboard interface).

Rudy Biesma - Betuwe 18
9405 JJ ASSEN - Niederlande
rudy@amazed.nl

```
; * TORNADO *  Rudy Biesma 2001

;Colour screendump routine for Canon BJC4300
;should work for other Epson (ESC/P) compatible
;printers

        ORG     50000
        DUMP    50000

start   LD      A,3
        CALL    #1601,CHAN_OPEN
                                ;open stream #3
        LD      HL,16384        ;screen start
line    PUSH    HL              ;character row
                                ;address
        LD      DE,kbuf         ;printer buffer
                                ;address
        LD      C,32            ;spectrum screen
                                ;has 32 columns
        LD      B,C
charcol LD      C,B
        LD      B,8             ;8 pixel rows per
                                ;character
charrow PUSH    BC
        PUSH    HL
        LD      B,8             ;8 pixels pro Byte
pixel   RLC     (HL)            ;carry set=ink pixel
                                ;carry not set=paper
                                ;pixel
        PUSH    HL              ;save pixel address
        PUSH    AF              ;save carry flag
        LD      A,H             ;calculate attribute
                                ;address
        RRCA
        RRCA
        RRCA
        AND     #03
        OR      #58
        LD      H,A
        POP     AF              ;restore carry flag
        LD      A,(HL)          ;get attribute
        POP     HL              ;restore pixel
                                ;address
        INC     H               ;next pixel row
        JR      C,colour        ;jump if ink colour
        RRCA                    ;get paper colour to
                                ;3 lsb
        RRCA
        RRCA
colour  AND     #07             ;A now holds pixel #
                                ;colour
        SCF                     ;signal black pixel
        JR      Z,shift         ;jump if black pixel
        XOR     #07             ;make CMY colours,
                                ;reset carry!
                                ;A=0 if white pixel
shift   EX      DE,HL           ;DE=next pixelrow,
                                ;HL=print buffer
        PUSH    HL
```

```
        RL      (HL)            ;shift black part into
                                ;buffer
        INC     H               ;next buffer
        RRA                     ;shift yellow bit into
                                ;carry
        RL      (HL)            ;shift into yellow
                                ;buffer
        INC     H               ;next buffer
        RRA                     ;shift cyan bit into
                                ;carry
        RL      (HL)            ;shift into cyan buffer
        INC     H               ;next buffer
        RRA                     ;shift magenta bit
                                ;into carry
        RL      (HL)            ;shift into maganta
                                ;buffer
        POP     HL              ;restore printer buffer
        EX      DE,HL           ;DE=printer buffer,
                                ;HL=next pixelrow
        DJNZ    pixel           ;loop for all 8 pixels
                                ;in byte
        INC     DE              ;next printer column
        POP     HL              ;restore pixel
                                ;address
        POP     BC              ;restore pixel
                                ;row & char.
                                ;column
        DJNZ    charrow         ;loop for 8 pixelrows/
                                ;char.
        INC     HL              ;next character
                                ;column
        LD      B,C             ;column to B
        DJNZ    charcol         ;loop for 32 columns
        XOR     A               ;A is colour number
        LD      HL,kbuf         ;Printer buffer
print1  PUSH    AF
        PUSH    HL
        LD      DE,table        ;Print message A.
        CALL    #0C0A,PO_MSG
        LD      DE,table        ;Set printer bit image
                                ;mode
        LD      A,4
        CALL    #0C0A
        POP     HL
        PUSH    HL
        LD      B,0             ;256 pixels
print2  PUSH    BC
        PUSH    HL
        LD      A,(HL)          ;send pixels to
                                ;printer
        RST     #10
        POP     HL
        POP     BC
        INC     HL
        DJNZ    print2          ;loop for 256 pixels
        POP     HL
        INC     H               ;next printer buffer
        POP     AF
        INC     A               ;next colour
        CP      4
        JR      NZ,print1       ;loop for 4 colours
        LD      A,5             ;new line
        LD      DE,table
        CALL    #0C0A
        POP     HL
        LD      A,H             ;calculate next
                                ;screen address
        ADD     8
        LD      H,A
        LD      A,L
        ADD     32
        LD      L,A
        CCF
        SBC     A
        AND     #F8
        ADD     H
        LD      H,A
        CP      #58
        JR      NZ,line         ;loop for all screen
                                ;rows
        RET

table   DEFB    #80             ;start marker
        DEFB    13,27,"r",0+128 ;Message 0, black ink
        DEFB    13,27,"r",4+128 ;M. 1, yellow ink
        DEFB    13,27,"r",2+128 ;M. 2, cyan ink
        DEFB    13,27,"r",1+128 ;M. 3, magenta ink
        DEFB    27,"K",0,1+128  ;M. 4, bit image
                                ;60dpi,
                                ;    256 bytes follow
        DEFB    27,"J",24+128   ;M. 5, 24/180 inch
                                ;linefeed

kbuf    DEFS    256             ;black buffer
ybuf    DEFS    256             ;yellow buffer
cbuf    DEFS    256             ;cyan buffer
mbuf    DEFS    256             ;magenta buffer
```